For Tandy's 100 and 200 Portables and 1000, 1200 and 2000 MS-DOS Computers

PCM®

The Personal Computing Magazine
for Tandy® Computer Users

Vol. III No. 1
July 1985
U.S. $3
CDN $3.95

Launching Our Third Year

— with —

A Complete Index to PCM

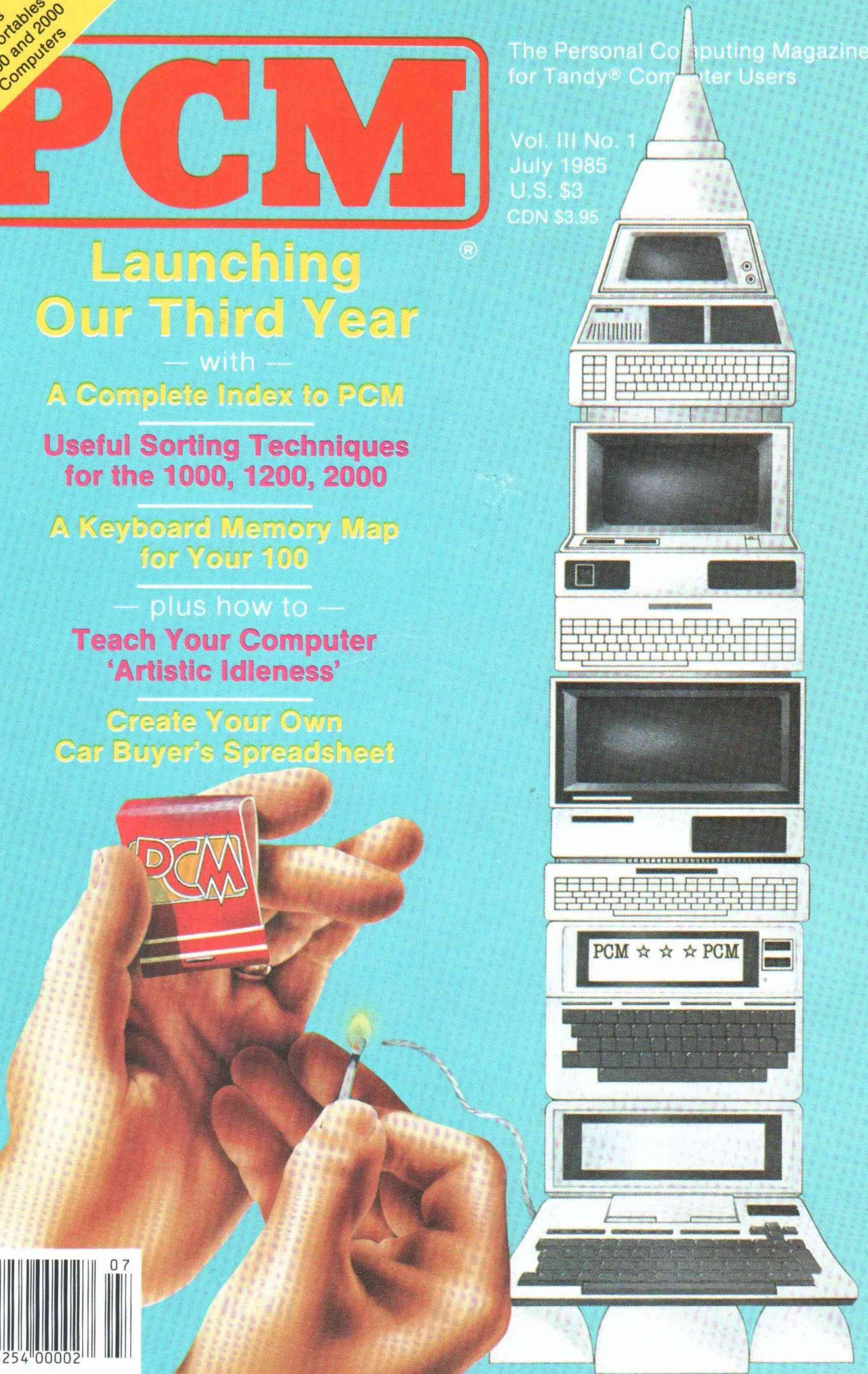
Useful Sorting Techniques for the 1000, 1200, 2000

A Keyboard Memory Map for Your 100

— plus how to —

Teach Your Computer 'Artistic Idleness'

Create Your Own Car Buyer's Spreadsheet



SO SIMPLE

TELECOMMUTER™

INTEGRATED WORD PROCESSING AND COMMUNICATIONS SOFTWARE

For the Tandy 1000, 1200 and 2000, and IBM-PC/XT/AT/jr and compatible computers. Uses the same editing and communications commands as the Model 100/200 — plus much, much more!

EASY TO USE . . .

"I am writing to let you know that I like your program very much indeed. The high speed transfers from my Model 100 to my Tandy 1000 are flawless. So many word processors are encumbered by in my view unnecessary baggage for the normal user. Yours is powerful, yet the user interface is elegant in its simplicity. It is essentially a 'Super Model 100'. Your advertising does not express the versatility of the program. I was intending to use it for general writing and articles. I now intend to use it for a book I'm working on."

Jay S. Moynihan, Esq.
Ashland, WI

WITH POWER FOR THE BIG JOBS!

"A few weeks ago I received Telecommuter . . . it has been in constant use as a word processor, a telecommunications package (which has allowed me to access numerous databases), and to transfer files from my Model 100 to the Tandy 2000. Since Telecommuter has arrived, both Word Perfect and Videotex Plus . . . the word processor and terminal program I had been using . . . have not left the shelf. I have also used Telecommuter to write dBase II command files, and it has made that job so much easier than when utilizing the programming section of dBase itself. My most enthusiastic congratulations on a superior product."

Ivan K. Goldberg, M.D.
New York, NY

Available in six versions:

TC-Write It™ $125, **TC-Standard** and **TC-XMODEM** $200, **TC-DeLuxe** $300, **TC-plus** $400, **TC-Corporate** $1000.

Advanced features include DOS access from Word Processor, XMODEM file transfer, Telcom macros, and full VT100 terminal emulation. Host Mode gives control of your computer to a remote Model 100/200 or MS-DOS computer running Telecommuter.

Call or write today for more information on the entire Telecommuter product line. Product upgrades available for difference in price. We accept MasterCard and Visa for mail or telephone orders, and we ship free in the USA.

Look for Telecommuter at your local Radio Shack Computer Store!

SIGEA SYSTEMS, INC.

19 Pelham Road, Weston, MA 02193 (617) 647-1099

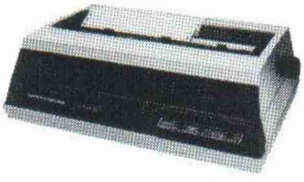
From Computer Plus to YOU . . .

PLUS *after* PLUS *after* PLUS

Tandy 200 24K $ 739
Model 100 8K $339
Model 100 24K $425

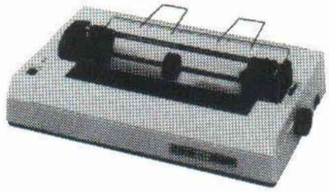
Color Computer II
w/16K Ext. Basic $135
w/64K Ext. Basic $179

Tandy 1000 $779
Tandy 1200 $1599



Smith Corona L-1000
$259

Color Computer Disk Drive
Drive 0 $279 Drive 1 $169



DMP105 $160

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Tandy 1000 1 Drive 128K	779.00
Tandy 1200 10 Meg. 256K	1599.00
Tandy 2000 2 Drive 256K	1599.00
Tandy 2000 10 Meg. 256K	2750.00
Model IV Portable 64K	970.00
Model IV Desktop 64K	959.00
PRINTERS	
Radio Shack DMP-110	299.00
Radio Shack DMP-430	660.00
Radio Shack DWP-210 Daisy Wheel489.00	
Radio Shack TRP-100 Portable	229.00
Silver Reed EXP-550 Daisy Wheel	430.00
Star Powertype Daisy Wheel	345.00
Star SG-10	245.00
Star SG-15	410.00
Star SD-10	365.00
Panasonic P-1091	275.00
CITOH Prowriter 8510AP	320.00
CITHO 7500	230.00
Okidata 92	389.00
Okidata 93	625.00
Epson	CALL

Tandy 1000/1200 ACCESSORIES	
Tandy 1000 Disk Drive Kit	159.00
256K Ram Board (inc. 128K & DMA)	199.00
512K Ram Board (includes 128K)	169.00
128K Ram Upgrade Kit (Tandy)	135.00
128K Ram Upgrade Kit (NEC)	98.00
300 Baud Modem Board	129.00
RS-232 Serial Board	89.00
Hard Disk Controller Board	249.00
Digi-Mouse/Clock Board	89.00
Tandy VM-2 Green Monitor	129.00
Tandy CM-2 RGB Color Monitor	379.00
Tandy VM-3 TTL Green Monitor	179.00
Tandy 1200 Text Monitor Adapt.	179.00
Tandy 1200 Color/Graphics Adapt240.00	
MODEMS	
Radio Shack DCM-3	52.00
Radio Shack DC Modem IB	89.00
Radio Shack DC Modem II	160.00
Radio Shack DC Modem 2212	315.00
Hayes Smartmodem II 300 Baud	215.00
Hayes Smartmodem 1200 Baud	429.00
Novation J-Cat 300 Baud	115.00

MODEL III/IV ACCESSORIES	
PTC-64K Printer Buffer	199.00
Model III/IV Drive 0 Kit	465.00
Model III/IV Drive 1 Kit	220.00
Model III/IV Hi-Res Board	229.00
Model III/IV RS-232 Board	99.00
Model IV 64K Upgrade Kit	79.00
FOR EVERYONE	
Radio Shack Powerline Filter	45.00
CCR-81 Cassette Recorder	52.00
CCR-82 Cassette Recorder	43.00
C-20 Digital Cassette Tapes (10pk)	9.95
Verbatim SSDD Plastic Box (10pk)	24.95
Verbatim DDDD Plastic Box (10pk)	34.95
Generic SSDD Diskettes (10pk)	17.95
NEC 64K Ram Chips (set of 8)	49.00
Radio Shack 16K Ram Chips (8)	25.00
Locking Disk Box (holds 70)	19.95
Flip N File Disk 50	15.00
Fanfold CleanEdge Paper (2600)	35.00

Radio Shack software 10% off.
Send for complete listing of
brand name software and hardware.

CALL TOLL FREE
1-800-343-8124

- *LOWEST POSSIBLE PRICES*
- *BEST POSSIBLE WARRANTY*
- *KNOWLEDGEABLE SALES STAFF*
- *TIMELY DELIVERY*
- *SHOPPING CONVENIENCE*

VISA MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (617) 486-3193

TRS-80 is a registered trademark of Tandy Corp.

PCM
The Personal Computing Magazine for Tandy® Computer Users

Vol. III, Issue 1 July 1985

Editor and Publisher Lawrence C. Falk

Managing Editor Courtney Noe
Senior Editor Jim Reed
Technical Editor Danny Humphress
Submissions Editor Jutta Kapfhammer
Copy Editor Tamara Dunn
Reviews Editor Monica Dorth
Editorial Assistants Jody Doyle, Wendy Falk, Debbie Hartley, Judi Hutchinson, Angela Kapfhammer, Belinda Kirby, Suzanne Benish Kurowsky, Shirley Morgan, Kevin Nickols
Editorial Consultant Vince Lord
Contributing Editors William Barden, Jr., Randy Graham, Jim Hawk, Richard White
Art Director Jerry McKiernan
Assistant Art Director Sally Gellhaus
Designers Tracey Jones, Heidi Maxedon, Kevin Quiggins
Technical Assistant Ed Ellers
Advertising Coordinator Doris Taylor (502) 228-4492
Advertising Assistant Debbie Baxter
General Manager Patricia H. Hirsch
Assistant General Manager for Finance Donna Shuck
Bookkeeper Diane Moore
Advertising Accounts Beverly Taylor
Dealer Accounts Judy Quashnock
Administrative Assistant To The Publisher Marianne Booth
Manager of Public Relations Charles L. Springer
Director of Fulfillment Services Bonnie Shepard
Assistant Customer Service Manager Deidra Henry
Customer Service Representative Sandy Apple, Monica Wheat
Word Processor Manager Lynda Wilson
Research Assistants Judy Brashear, Laurie Falk, Sharon Smith
Dispatch Janice Eastburn
Production Assistant Melba Smith

For PCM Advertising and Marketing Office information, see Page 90.

PCM — The Personal Computing Magazine for Tandy® Computer Users is published every month of the year by FALSOFT, INC., 9529 U.S. Highway 42, P.O. Box 385, Prospect, KY, 40059. Phone (502) 228-4492. **PCM — The Personal Computing Magazine for Tandy® Computer Users** and the **PCM** logotypes are registered ® trademarks of FALSOFT, Inc.

Second class postage paid Prospect, KY, and additional offices. USPS N 713-470 (ISSN 0747-0460). POSTMASTER: Send address changes to PCM, P.O. Box 385, Prospect, KY 40059. Forwarding Postage Guaranteed.

Subscriptions to **PCM — The Personal Computing Magazine for Tandy® Computer Users** are $28 per year in the United States. Canadian rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only.

Limited back issues are available. Please see notice for issues which are in print and costs. Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in U.S. currency only.

PCM Contents

Features

Departments

Reviews

Cover illustration by Fred Crawford copyright© 1985 by Falsoft, Inc.

Tandy computers offer everything . . . almost

Check out the software that completes their systems —*filePro*®

Tandy MS-DOS Software Comparison Chart

	pfs® File/Report	dBase™ II	*filePro* 16 Smallware
GENERAL CHARACTERISTICS:			
Menu driven	yes	no	YES ✓
Allows user to create integrated business systems	no	programmer required	YES ✓
Developed systems and data can be moved to multi-user environments	no	no	YES ✓
Professional support available from the software's authors	no	no	YES ✓
PRICE	$265	$595	$495
CAPACITIES:			
Fields per record	100	32	999 ✓
Characters per record	1679	1000	4608 ✓
Records per file	1300	65535	16,000,000 ✓
Indexes per file	1	7	12 ✓
Number of digits per numeric field	20	10	24 ✓
Number of files usable concurrently	1	2	10 ✓
Files span multiple drives	no	no	up to 8 ✓
FEATURES:			
Full-screen facility for creating custom screen layouts	yes	no	YES ✓
Full-screen facility for creating custom report layouts	no	no	YES ✓
Built-in field types (error checking)	no	3	12 ✓
User-defined field types	no	programmer required	200 ✓
Conditional math	no	programmer required	YES ✓
User-defined menus	no	programmer required	YES ✓
Change file layout without losing existing data	possible	possible	automatic ✓
Data protection	no	programmer required	YES ✓
Password security	no	programmer required	YES ✓

The new Tandy computers can really speed things up. If you don't let obsolete software bog you down.

Try *filePro* 16, the latest generation of data base software from the authors of Profile®.

Choose it for its power (check our capacities); its ease-of-use (check our features); and its unique ability to grow as your needs grow, and even move the systems you create to multi-user computers.

filePro is available for the Tandy 1000, 1200 and 2000 computers (hard disk required). Available directly from us or through your Radio Shack store's Express Order Software service.

To order, call (800) 847-4740.

For customer service or when dialing in New York State, call **(212) 398-9290.** Ask for Smallware sales. Visa and MasterCard accepted.

Also available for other computers, including IBM PC, IBM PC-XT, and NCR Decision Mate.

The Small Computer Company, Inc.

240

230 West 41st Street, Suite 1200, New York, New York 10036

Smallware and *filePro* are registered trademarks of The Small Computer Company, Inc. dBase II is a trademark of Ashton-Tate. pfs File/Report is a registered trademark of Software Publishing. Profile is a registered trademark of Tandy Corporation.

A Most Healthy and Confident Two Year Old

It is hard to believe we are now starting our third year of publishing PCM. Yet, we're really pleased that we're stronger and more of a force in the marketplace than at any time in our history. For this our thanks go to you, our loyal readers and subscribers, who have been our main supporters through these years.

PCM has changed quite a bit since the first issue rolled off the presses. We now cover all the major Tandy computers except the Color Computer, to which our sister publication, THE RAINBOW, is devoted. And, we've expanded in such a way that we feel very confident we will be a part of the marketplace for years to come.

That is no small statement. A recent *Time* magazine article pointed out that there were some 150 computer magazines two years ago and there are but 40 today. Since we have a total of three computer magazines (the other is SOFT SECTOR for the Sanyo line), I suppose that means we have a total of eight percent of the market. All three are healthy, I might add.

One of the things that has been so gratifying about your following PCM is that many of you have so graciously mentioned our name when you order products from our advertisers. In fact, several of you have mentioned us when you order things from non-advertisers as well. We've picked up a couple of ads that way, and it really helps us (and you, too, because it helps us grow).

I know the old saying "please patronize our advertisers" sounds a bit trite, but it is important to our success. Many thanks for all the help you've given us so far. And, too, thanks for telling so many others about PCM — and, for that matter, the "clearly superior" line of Tandy computers.

We're the *only* magazine anywhere which supports the Tandy MS-DOS machines: the 1000, 1200 and 2000. And we're one of only two which supports the 100 and 200 portables. We believe we've carved a nice and growing niche for ourselves and for you, and we're looking forward to continuing.

So as year three dawns, thank you for supporting us. Allow me to echo many of *your* letters to us and say "Keep up the good work." As we continue to grow and expand, your support means a great deal to everyone here at PCM.

* * *

You will recall I mentioned the possibility of a PCMfest show for the fall and asked for comments. The response to this idea has really been overwhelming, and we are going ahead with plans for a show.

The date is Oct. 11-15 at the Hyatt-Regency Princeton in Princeton, N.J. —about equidistant from New York City and Philadelphia. Newark airport is the best way to fly in.

More details are coming and there will be an order form for tickets. As I said before, the show will be in conjunction with our highly successful RAINBOWfest for the Color Computer, but there will be a separate display area just for Tandy MS-DOS and Portable computers. Still, your ticket will get you in to the Color Computer show, too.

We're also planning a series of seminars and the tickets will be good for those as well, so mark your calendar now and plan to attend. I think it will be a very enjoyable, entertaining and stimulating weekend. Tickets are $7 for one day or $11 for three days, if purchased in advance. At the door, the tickets are $9 and $13. There is also a special PCMfest rate available from the Hyatt — but it is a good idea to make reservations early since the hotels tend to sell out.

* * *

Who likes the Tandy 1000? *Infoworld*, according to a review which appears in a recent issue under the headline "Tandy's Magnificent Concession." The 1000 draws a rave review with a summary which says it is "a valuable business machine and home computer with impressive potential for expansion in any direction."

What *Infoworld* does not say is that there are some very positive changes coming from Tandy for the 1000 as well. Those, plus some very nice options from third party suppliers, means this sweet little machine is pretty much *the* computer to have — performance, price and serviceability make it so.

And, one of the truly impressive things about the 1000 is something else new from Tandy — its Vianet networking system. With some assistance from the people in Fort Worth, we have been delving into Vianet and plan a major takeout on it next month.

I don't want to usurp all of Technical Editor Danny Humphress' surprises, but what Vianet appears to be able to do is link literally dozens of MS-DOS computers (1000s, 1200s, 2000s and even IBM PCs) together into one system. That means you can share files and abilities between computers and, even more important, peripherals! You could, for example, hook up one very high-speed printer to one computer and let everyone use it. Or, in a more interesting application, you could load three printers with three different types of letterhead and then just send letters to the printer which has the letterhead you want to use.

Our first look-see on Vianet impresses us a whole lot. In the next couple of weeks we will get into it in depth. We even expect to include some programs you can use with a Vianet system!

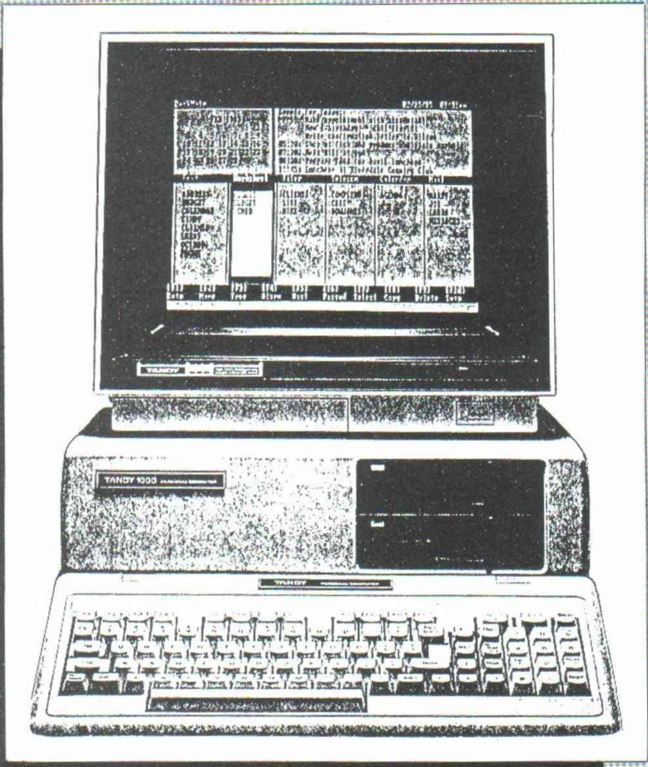
— Lonnie Falk

TANDY 1000

INTERNAL *HARD DRIVES*

- NO ADDED MEMORY BOARD NECESSARY
- BOOTS OFF HARD DRIVE
- TERMS—
  MONEY ORDER
  CASHIER'S CHECK
  *C.O.D.
  *MASTERCARD/VISA
  5% FEE

*PLUS SHIPPING & HANDLING



- COMPLETE KIT— NOTHING ELSE TO BUY
- INSTALLS IN MINUTES WITH ONLY A SCREWDRIVER
- INTERNAL KITS AVAILABLE FOR*:
  TELEVIDEO
  TANDY 1000
  SANYO
  COMPAQ
  LEADING EDGE
  PANASONIC
  CORONA
  IBM PC
  NOVA
  AT&T, NCR & ITT

10 MEG
$895.00

20 MEG
$1195.00

32 MEG
$1395.00

HARD DRIVE & CONTROLLER ARE 100% IBM COMPATIBLE. WE USE ONLY QUALITY DRIVES & WESTERN DIGITAL CONTROLLERS FOR THE HIGHEST QUALITY. WE BACK THE FULL SYSTEM WITH A 1 YEAR WARRANTY. THE SYSTEM COMES COMPLETE WITH ALL CABLES & MOUNTING HARDWARE. REQUIRES NO ADDED MEMORY BOARD TO OPERATE. 20 & 32 MEGABYTE INTERNAL HARD DISKS ALSO AVAILABLE. COMPLETE INFO AVAILABLE UPON REQUEST.

RTX TECHNOLOGIES, INC.
P.O. BOX 50398
NASHVILLE, TENN. 37209
(615) 373-1251

*SAME PRICE AS TANDY
*ALL MS™-DOS COMPUTERS

WATCH THOSE MAXFILES

Editor:

Thank you for an excellent publication. Your magazine provides many useful tips and programs for use on my Model 100. Maybe you can help with a problem I have been having with file entries.

The 100 sporadically gives me "BN" error messages when my BASIC program calls for file handling. At first I thought it was a software problem and kept trying to change the program — with little success. Then I noticed that it happened to programs and files that had worked in the past.

The local Radio Shack service center generously changed out the entire logic board because it did appear to be a hardware problem. It worked fine in the shop, but came up with the same BN error as soon as it was back in my office, only to dissappear again that same afternoon. Now it is back, but on a different program, while I am travelling.

I have tried changing batteries, power supplies and the logic board. It is a 16K machine and the second 8K memory chip was not changed when the logic board was replaced. They also used the same NiCad battery. Could the problem be in one of those components?

Chuck Stapleman
Glendale, WI

Editor's Note: More than likely, the problems you have been experiencing are related to the idiosyncracies of the MAXFILES *command. This is fortunate, because the cure for all the trouble it has caused you is quick and simple. If one of the programs you have been having trouble with is our bar code reader BR2.1, change Line 1020 of the reader program to the following:*

```
1020 CLEAR 1000: MAXFILES = 2
```

To ensure that a program will run properly if it is necessary to open RAM files, you should reassert the MAXFILES *value, making it equal to the number of files you will need to open. See the following letter.*

A PHENOMENAL STATEMENT

Editor:

Did you know that the MAXFILES command [in the 100] neutralizes an earlier DIM statement? If not, try the following program.

```
100 DIM X(20)
110 X(17)= 17
120 PRINT X(17)
```

This program will print 17. Now insert the following line:

```
105 MAXFILES = 2
```

When you run this program you will get a "BS Error in 110." Tests will show that a PRINT X(10) will return 0, but PRINT X(11) will return a "BS Error." This idiosyncrasy is not mentioned in the manual.

P.O. Hanson, Jr.
Largo, FL

Editor's Note: See Aileen and John Cornman's article on "What's Not in the Model 100 Manual," PCM, October 1984, Page 20.

RESPONDING IN KIND

Editor:

I would like to thank all of the nice people who responded to our call for new members. The response was very good. Our membership at this time has jumped by 25 percent!

I would also like to say that we have just started another public domain software base. This now totals three public domain software bases.

We are a club just for the Tandy 2000 and we are always looking for new members. (2000 Land Microcomputer Club, 6522 Clara St., 90201)

Lance Gish
Bel Gardens, CA

MAKING THE SWITCH

Editor:

I recently purchased a Tandy 1000 and I had my dealer call and become a distributor of your publication. Having been a Color Computer owner, I am well aware of the difference a support publication can make in enjoying and getting the most from a computer system. THE RAINBOW will be tough to top!

Information on the Tandy 1000 seems scarce, at best. A usual malady associated with being a newcomer to the industry. I am seeking a source for a high resolution graphics screen dump for my DMP-200 printer. It seems that most of the software houses only support Epson and look-a-likes in the IBM PC arena.

I assume other former CoCo users will be buying the Tandy 1000, 1200 or 2000. The readers may note that many of the programs from THE RAINBOW will run under GW-BASIC with few modifications. There are differences in the screen positions and subtle differences in the code, but by and large, the programs will work. Don't use POKEs!

Thanks, PCM, for coming to the rescue. I'm still getting used to the Tandy 1000 and I need all the help I can get!

Gary Hawkins
Ladysmith, WI

THE WHOLE STORY

Editor:

Since the start of my subscription, there's been something about your magazine that set it apart in my mind from the others. I hope I'm not jumping to an assumption, and what I note has not occurred by chance.

PCM seems to have an editorial policy to keep the articles intact, whereas the other magazines I read chop them up and make the reader hunt through the magazine for its pieces. By keeping them intact, you make the magazine more readable. And, that's a very positive, but subtle, difference.

Ronald Balonis
Trucksville, PA

Editor's Note: You're not mistaken. We think it makes PCM more readable, too.

ODE TO MODEL 100 USERS

Editor:

Please allow me, as a 2000 owner, to address a segment of your readership and ask their forebearance: Dear Model 100 faithful, I ask for at least a state of peaceful co-existence in PCM. Let me explain why I am a subscriber to a magazine that you have regarded as your own home turf before we 2000'ers interloped.

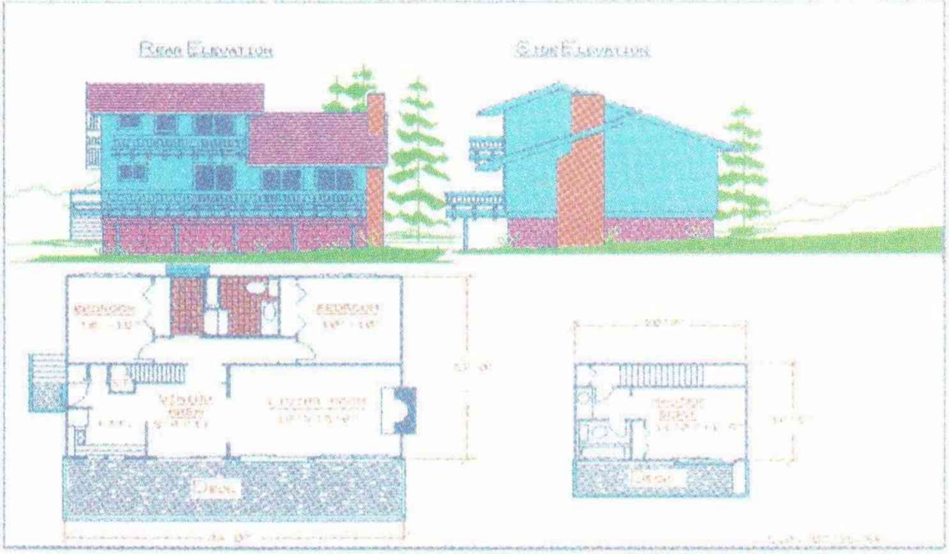
I subscribed to PCM as soon as I found out that it would cover the Model 2000 for two reasons. Number one, we 2000'ers have no other life support system. *80 Micro* pitched us out in the cold just like they did you. All the other decent magazines on the market can't see anything but the three magic letters — IBM. The 2000 is a great machine, and while it can do three things while the PC is thinking about doing one, it just is not a PC clone, and very little published for the IBM PC is usable by the 2000 without modification of some type. We need support because we are few enough in number that no after-market group has built up around the 2000 like it has for every other Tandy machine. As an example, we have absolutely no source for add-on boards and hard disk systems but Tandy, and you know what they charge.

The other reason for subscribing specifically to PCM is that Lonnie Falk and Falsoft are survivors. I wasn't a charter subscriber to THE RAINBOW. Two or three issues had been published before I found out about it. I was, however, a charter subscriber to the other three Color Computer magazines, and two of the three are no more. The third magazine is smaller than PCM and THE RAINBOW looks like a first cousin to *Byte*. If this serves as an example, PCM is going to get better and better, and is going to be around for a long time. I hope we can get along in the same publication.

After all, I use my portable as an adjunct to my 2000, and the two have become almost inseparable friends. My two primary uses of my portable are to produce *WordStar* files and Turbo PASCAL source files that I can download to my 2000 when I get home. The Model 100 is great for this kind of application. Please . . . peace.

David M. Dacus
Las Cruces, NM

Leonard Guay's drawing using *Skedit*.



IDEAL COVERAGE

Editor:

As an owner of a Model 100 and Model 2000, I think your magazine is ideal. Sorry to see so many letters to the editor requesting no 2000 coverage. Keep up the good work.

Leon Morrison
Morehead City, NC

DOTS AND LINES

Editor:

I thoroughly enjoy reading PCM and look forward to receiving each monthly issue. Thank you for your support of the Tandy 2000. Having purchased one of the first units available in the Jacksonville area, I have spent many hours developing graphics programs for my personal engineering applications and *CAD* program. The *Skedit* and *Scomp* programs by Dave Pifer, published in your recent issues, proved to be of tremendous help. In my estimation, this is the best graphics program ever published in any computer magazine.

In the February issue of PCM, Jeff Helms described a problem with dots or short lines appearing on the screen when idled for a fairly long period of time. This is a problem I have also encountered with my equipment since the date of purchase. Any discussions of this malfunction with the personnel at the local computer center have resulted in vague suggestions as to the computer being overloaded and overheating. Have any of your staff experienced this difficulty with any of their systems? Are there any corrective measures that can be taken without returning the computer to the Tandy service department?

Thanks again for your comprehensive coverage and software for all Tandy computers. Tandy really owes PCM for this invaluable support.

Leonard Guay
Jacksonville Beach, FL

Editor's Note: This is a fairly common problem with earlier Tandy 2000 color systems. When we experienced the phenomenon here, it was corrected with a quick trip to the local Tandy service center. Have the people at your local computer center get you in touch with their local ATSO (Area Training and Service Office), or the nearest repair facility. They should be able to take care of the problem.

Using Machine Language Subroutines in BASIC

By Robert D. Covington

Have you ever wished you could sort a 1000 element array in less than three seconds? Have you ever wished you could use BASIC to write your own command processor? Have you ever needed all the data found on a full-sized directory loaded into a string in BASIC?

If so, what do you do? One solution is to learn everything you can about machine language from the MS-Assembler manual and try to figure out how to write machine language programs for BASIC. The other and simpler alternative is to buy the next few months of PCM and read this column.

In the next few installments of "Subroutine City," I will describe quite a few machine language subroutines that extend the flexibility and speed of standard GW-BASIC. For those of you who do not know very much about machine language programming, don't panic! None of the machine language subroutines described in the next few installments require very much knowledge of machine language to use. As a matter of fact, if you are not interested at all in learning how machine language subroutines interface with BASIC, you can probably skip the sections entitled "Interfacing Machine Language Programs to BASIC" and "Installing Machine Language Programs in BASIC."

Machine Language in BASIC

When Microsoft designed GW-BASIC, they included two methods for accessing machine language subroutines. The first method uses the DEFUSRn and USRn( ) instructions. These instructions allow up to 10 machine language subroutines to be accessed inside a single BASIC program. The DEFUSRn statement is used to define the address where BASIC calls a machine language subroutine. The 'n' in DEFUSRn is a number from 0-9 determining which of the 10 USR functions is being defined. The USRn( ) function is used to call the machine language subroutine defined by DEFUSRn. With USRn( ) the value inside the parentheses can be easily passed to a machine language subroutine. In addition, since USRn( ) is a function, the variable at the left of the equal sign can receive a returned value from the machine language subroutine. For example, if BASIC executed a X=USR1(12), the routine at the address defined by DEFUSR1 would be called, a 12 would be passed to the machine language subroutine, and the value returned from the subroutine would be loaded into X. These features allow a machine language subroutine to communicate easily to BASIC programs.

The second method for accessing machine language subroutines in GW-BASIC uses the CALL instruction. This instruction allows an almost unlimited number of machine language subroutines to be accessed in a single BASIC program. With CALL, the first variable parameter defines where BASIC should call the machine language subroutine. For example, if an:

```
A=65000:CALL A
```

were executed in BASIC, the routine at 65000 in BASIC's current segment would be called. For some reason, BASIC requires that all parameters used by CALL be variables. This means that the above example could not be replaced with a CALL 65000. Following the transfer address variable, any number of variables can be listed inside a pair of parentheses to be accessible by the machine language subroutine. For example, if an:

```
A=65000:CALL A(A$,X,D,Y%)
```

were executed in BASIC, the routine at 65000 would be called with A$, X, D, and Y% transferring data to or from the machine language subroutine. Of course, the variables must be the same type and be in the same order expected by the machine language subroutine.

In most cases, the CALL instruction is usually preferred over the USR( ) function. One reason for this is that the CALL instruction does not limit how many parameters can be passed like the USR function. In addition, unlike CALL, the USR( ) function limits the number of machine language subroutines inside a single BASIC program to 10. As a matter of fact, the only reason the USR( ) function was implemented in GW-BASIC was to keep some compatibility with older Microsoft BASICs.

Since USR and DEFUSR are rather obsolete instructions, I will not be using them in any of the machine language subroutines presented in this column.

Interfacing CALL to ML Programs

Interfacing a machine language program to BASIC's environment is quite easy when using CALL. When BASIC encounters a CALL instruction with no variable parameters (example, CALL A), all of the CPU's registers are saved on the stack and a long call to the machine language subroutine is performed. The address for the long call is

Prices Change
Every Day.
Please Call
1-800-343-8841
For Lower Prices.

PRICE BREAKTHROUGH

Prices Change
Every Day.
Please Call
1-800-343-8841
For Lower Prices.

DEALER INQUIRIES INVITED

MEGADISK™ HARD DISK DRIVE SYSTEMS

TOLL FREE ORDERING 1-800-343-8841

For the { **IBM/PC**, Tandy 1000, **TRS/80 Models I/III/IV/4P**, Compaq, Eagle, Sanyo,
Tava, PC Workalikes, **Color Computers**, Heath/Zenith, Max/80
Complete with Hardware, Cables, Software and Quikfit Installation

5	Megabytes Internal Mount	**starting at**	**$ 299.95**
10	Megabytes Internal Mount	**starting at**	**544.95**
20	Megabytes Internal Mount	**starting at**	**799.95**
5	Megabytes External System	**starting at**	**499.95**
10	Megabytes External System	**starting at**	**699.95**
20	Megabytes External System	**starting at**	**999.95**
	Streaming Tape Backup System — Internal Or External	**starting at**	**499.95**

Wow! New Low Prices

DOS Systems Available: { IBM/Heath — DOS, 1.0, 2.0, 2.1, 3.0, or later
TRS/80-LDOS, TRSDOS 6.x, Newdos/80, Dosplus 3.5, CP/M, COCO DOS, Max/80 LDOS, OS9

FULLY WARRANTIED — PARTS AND LABOR — 24 HOUR SERVICE — CALL TOLL FREE — 1-800-343-8841

FLOPPY DISK DRIVES, POWER SUPPLIES AND CABINETS

Our Disk Drives are UL approved — Our Floppy Drive Cabinets and Power Supplies are Underwriters Laboratory Listed and have passed the required Federal Communications Part 15 Section B-EMI/RFI tests.
Warranty on all disk drives is one full year parts and labor. Warranty on floppy disk drive power supplies is five (5) years. In warranty or out of warranty service is 24 hour turn-a-round on all disk drives and power supplies.

Tandon — Full Height Drives		
100-1	Single Sided 40 tk Bare	$125.00
	In Case with Power Supply	166.95
	Dual Drives in One Cabinet	326.95
100-2	Dual Sided 40 tk Bare	162.00
	In Case with Power Supply	206.95
	Dual Drives in One Cabinet	381.95
	Half High Drives	
65-1	Single Sided 40 tk Bare	122.00
	In Case with Power Supply	156.95
	Dual Drives in One Cabinet	306.95
65-2	**Dual Sided 40 tk Bare**	**142.00**
	In Case with Power Supply	**176.95**
	Dual Drives in One Cabinet	**346.95**
65-4	Dual Sided 80 tk Bare	162.00
	In Case with Power Supply	196.95
	Dual Drives in One Cabinet	356.95
TEAC Half High Drives		
55A	Single Sided 40 tk Bare	125.00
	In Case with Power Supply	166.95
	Dual Drives in One Cabinet	316.95
55B	Dual Sided 40 tk Bare	157.00
	In Case with Power Supply	196.95
	Dual Drives in One Cabinet	356.95
55F	Dual Sided 80 tk Bare	177.00
	In Case with Power Supply	216.95
	Dual Drives in One Cabinet	386.95
Apple/Franklin Disk Drives		
	35/40 Track in Case with Cable and Software	159.95
	Controller Card for Two Disk Drives	46.95
	Combination Price for Disk Drive and Controller	196.95
Commodore Disk Drives		236.95
Power Supplies and Cabinets 5¼" and Hard Drive Systems		starting at 42.00

COLOR COMPUTER DISK DRIVE SYSTEMS AND ADD IN PRODUCTS

40 Track Single Head Drive with Case, Power Supply, Cable Controller, Instruction Booklet, Diskettes	$239.95
Above with Dual Drives in One Cabinet	374.95
40 Track Dual Head with Case, Power Supply, Cable, Controller, Instruction Booklet, Diskettes	**259.95**
Above with Dual Drives in One Cabinet	**399.95**
128 Memory Upgrade Kit	129.95
Dual DOS Switch	19.95
With Second DOS System — JDOS, RSDOS, Micro DOS, and Booklet	69.95

MODEMS

Volksmodem 300 Baud	$ 69.95
Signalman Mark X Autodial	123.95
Mark XII 1200/300 Baud Autodial	284.95

ALL IN-STOCK ITEMS SHIPPED WITHIN 24 HOURS. SAME DAY SHIPPING PROVIDED BY REQUEST WITHOUT ANY EXTRA HANDLING CHARGES.

************ MEGADISK SPECIAL ************

With a 10 megabyte system, you can purchase a complete IBM or PC compatible host, software and cable for only $15.00!!
P.S. Ask about the MEGAPLEXER.

IBM - PC/XT WORKALIKE

All of our computers have: 8 slot motherboard, 640K, monochrome adapter, parallel printer port, 130 watt power supply, free software — all with our full warranty.

/PC- 2 Floppy Drives - Monochrome Monitor, MS-DOS	$1,595.00
Color system resolution - 720h x 240v, MS-DOS	$1,995.00
5meg/**XT**tra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	$1,745.00
Color system resolution - 720h x 240v, MS-DOS	2,145.00
10meg/**XT**tra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	2,095.00
XTtra - Color system resolution - 720h x 240v, MS-DOS	2,495.00
20meg/**XT**tra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	2395.00
XTtra - Color system, MS-DOS	2,795.00
Internal Tape Backup For Any Of Above systems Add	$449.95

PRINTERS

Dot Matrix		
Citizen		$ Call
Star Micronics		
Gemini 120 CPS Near letter quality	starting at	$259.95
Delta 10/15 160 CPS	starting at	424.95
Radix 10/15 200 CPS	starting at	649.95
Panasonic 1090		249.95
Daisy Wheel		
Silver Reed 440 80 Column 12 CPS		315.95
550 132 Column 19 CPS		439.95
770 132 Column 36 CPS		895.00
Olympia 132 Column 14 CPS with Form and Tractor		399.95
Apple/Franklin Printer Interface w/Graphics and Cable		84.95
Printer Cables	starting at	19.95
Printer Paper — Microperf Edge 1000 Sheets		16.95

ELECTRICAL

Surge Protectors — Line Filters — SL Waber — 6 Outlets with Switch	$ 39.95
Uninterruptable Power Supplies	399.95

MISCELLANEOUS

Diskettes in 10 Pack	**from**	**$ 10.95**
Twoprint Switches	from	99.95
Disk Drive Cables	from	16.00
Maintenance Cleaning Kits		12.00
Parallel Printer Buffers 8K		149.95
Floppy Disk Drive Cables		
1 Drive		16.00
2 Drives		18.95
Heath/Zenith 2 Drive Cables — Shielded		24.95

SOFTWARE SUPPORT, INC.

1 Edgell Road, Framingham, MA 01701 (617) 872-9090 Telex-383425
Hours: Mon. thru Fri. 9:30 am to 5:30 pm (E.S.T.) Sat. 10 am to 3:30 pm

SERVICE POLICY — Our Professional Technical Staff Is Available To Assist You Monday Through Saturday.
WARRANTIES — Up To One Full Year Parts And Labor. Floppy Disk Drive Power Supplies — Five (5) Years.
SERVICE — 24 Hour Turn-A-Round On All In-Stock Parts. Dealer Inquiries Invited. Call 617-872-9090

Please Call For Shipping, Handling And Insurance.

Toll Free 1-800-343-8841

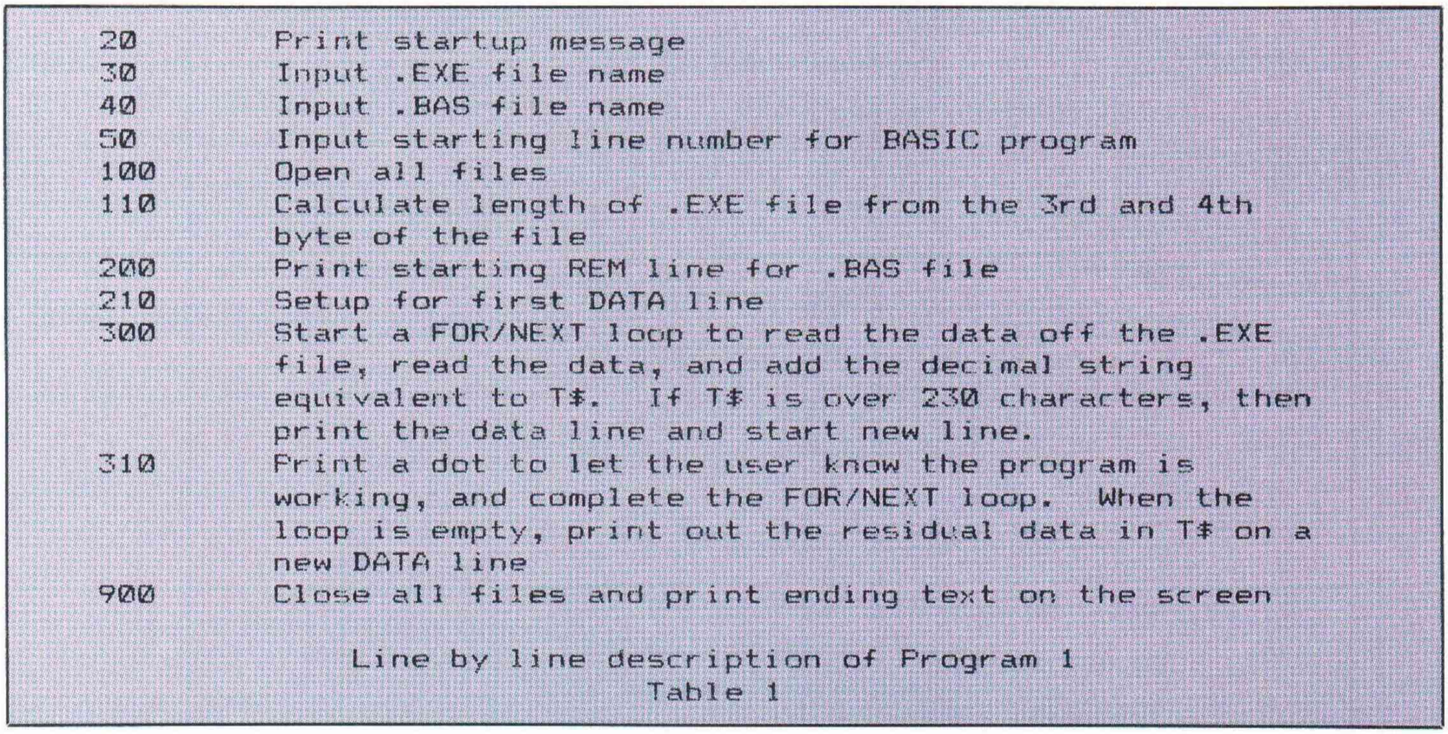
Please Call For Our Latest Price Saving Specials.

Not Responsible for Typographical Errors. Prices and Specifications May Change Without Notice.

derived from offset specified by the first variable in the CALL statement and the segment specified by the last DEFSEG instruction. If no DEFSEG instruction was ever executed in the program, BASIC's current data segment is used. Once the machine language program has performed its function, it should return back to BASIC with a far RETurn.

If the CALL instruction contains parameters (example, CALL A(A$,X)), BASIC processes the CALL slightly differently. Like a CALL with no parameters, all of the CPU's registers are first saved on the stack. Following this, BASIC stores a two-byte variable descriptor for each of the parameter variables on the stack. The variable descriptor stored is identical to the address returned from a VARPTR instruction. Finally, BASIC performs a long call to the machine language subroutine. When your machine language program receives control of the CPU, the stack resembles Figure 1.

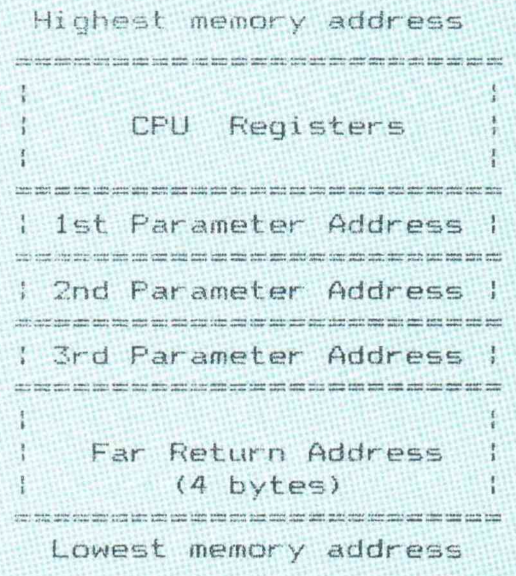
Once the machine language subroutine takes control, it will probably need to access the information on the stack about each of the parameter variables. The easiest way of doing this is the method described in Appendix E of the GW-BASIC reference manual. This method uses the BP (base pointer) register to read the two-byte variable descriptors off the stack. To do this, the machine language subroutine needs to load BP with the current position of the stack (MOV BP,SP). Then, using the 8088/186's direct, indexed memory addressing mode, each of the parameter descriptors can be accessed. In this addressing mode, the source memory is described as the contents of the BP register



```
   20        Print startup message
   30        Input .EXE file name
   40        Input .BAS file name
   50        Input starting line number for BASIC program
   100       Open all files
   110       Calculate length of .EXE file from the 3rd and 4th
             byte of the file
   200       Print starting REM line for .BAS file
   210       Setup for first DATA line
   300       Start a FOR/NEXT loop to read the data off the .EXE
             file, read the data, and add the decimal string
             equivalent to T$.  If T$ is over 230 characters, then
             print the data line and start new line.
   310       Print a dot to let the user know the program is
             working, and complete the FOR/NEXT loop.  When the
             loop is empty, print out the residual data in T$ on a
             new DATA line
   900       Close all files and print ending text on the screen
```

Line by line description of Program 1
Table 1

Condition of Stack on a 3 Parameter Call
Figure 1



```
  Highest memory address
==============================
!                            !
!       CPU  Registers       !
!                            !
==============================
! 1st Parameter Address      !
==============================
! 2nd Parameter Address      !
==============================
! 3rd Parameter Address      !
==============================
!                            !
!     Far Return Address     !
!         (4 bytes)          !
==============================
   Lowest memory address
```

added to a positive offset (example, MOV AX,[BP+4]). If you recall, Figure 1 showed that the bottom (lowest addresses) of the stack contains four bytes for the return address back to BASIC. Since most machine language subroutines do not need to know the return address, those four bytes should be skipped. Above the return address (SP+4) is the two-byte variable descriptor for the last parameter in the CALL. To access this parameter, MOVe the contents of BP+4 into a 16-bit register. To access the second to the last parameter, MOVe the contents of BP+6 into a 16-bit register. As a general rule, the positive offset of BP can be calculated by the formula:

2*(# of parameters-parameter position)+4

This formula assumes that nothing has been put on the stack by the machine language subroutine.

Once the machine language subroutine has easy access to the variable descriptors, the actual data in the parameter variables can be read. If the parameter described by the variable descriptor is a numeric variable, the two-byte descriptor points to the actual number inside of BASIC's data segment. In the case of an integer, the two-byte descriptor points to a two-byte signed integer. To retrieve the integer value, MOVe the contents of the variable descriptor into some register. For example, the following program would load the integer value of the last parameter of a CALL into the AX register:

```
MOV    BP,SP          ;Make BP base of stack
MOV    BX,[BP+4]      ;Get last parameter
                       descriptor
MOV    AX,[BX]        ;Load integer value into AX
```

With single and double precision numbers, the variable descriptor points to the four- or eight-byte floating point number. Appendix E of the GW-BASIC reference manual describes the format for floating point numbers. With strings, the variable descriptor points to a three-byte string descriptor table. The format of this table is:

Byte 0	Length of string
Byte 1	LSB of string address
Byte 2	MSB of string address

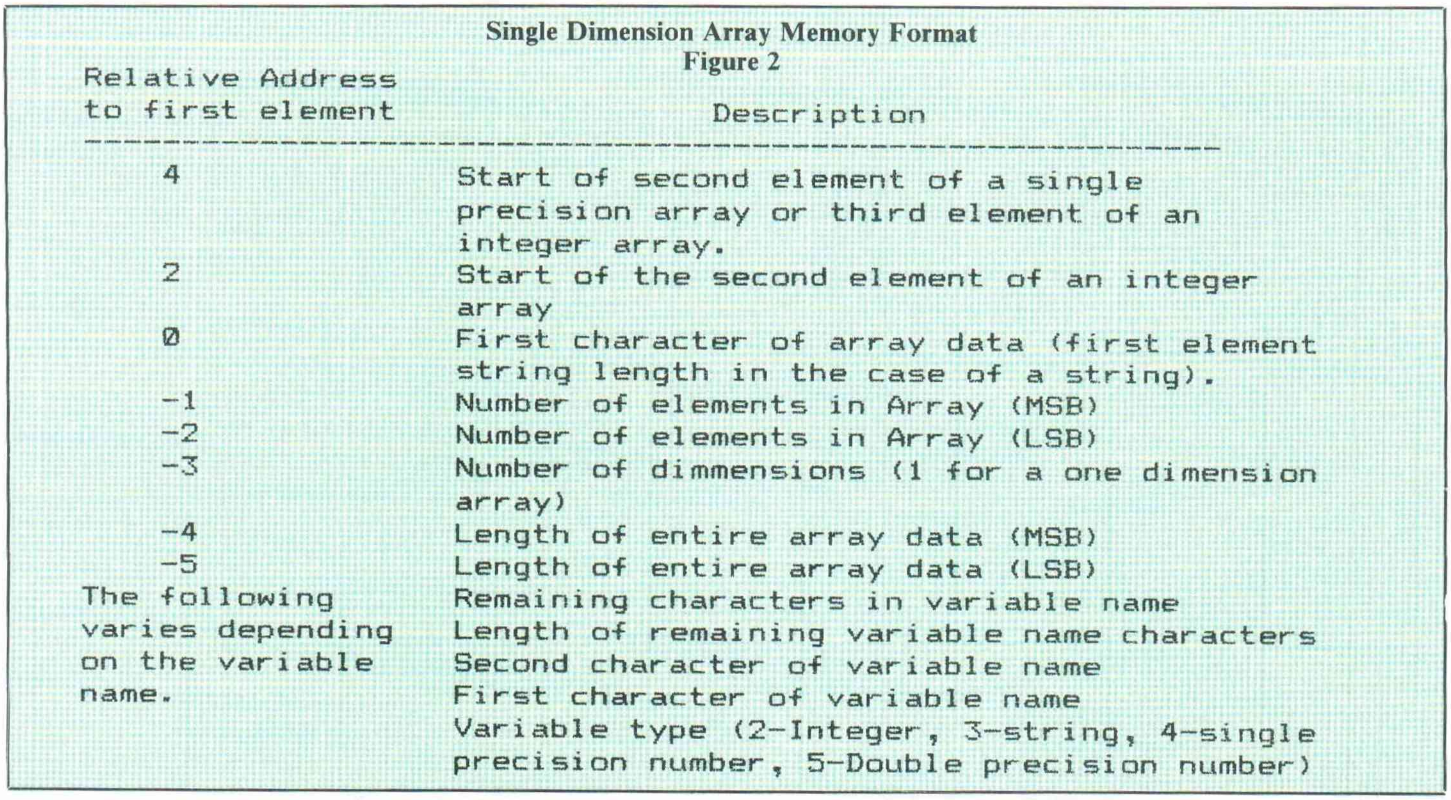
A program that would load CL with the length of the string and AX with the start address of the string would be:

```
MOV    BP,SP          ;Make BP base of stack
MOV    BX,[BP+4]      ;Get string variable
                       descriptor
MOV    CL,[BX]        ;Put string length in CL
MOV    AX,[BX+1]      ;Put string address in AX
```

If the parameter passed is an element from a string or numeric array, the variable descriptor points to the same information as a normal string or numeric variable.

As you can see, CALL allows information to be transferred from any string or numeric variable to your machine language program relatively easily. Unfortunately, CALL does not support accessing entire arrays as easily. This is unfortunate because one of the reasons for using machine language subroutines is to speed up array processing such as sorting, searching, etc. Luckily, however, there is a way to access information on an entire array with CALL. In GW-BASIC, the information preceding the first element of the array describes the entire variable. If you use CALL to find the variable descriptor address of the first element of the array, you can access enough information to manipulate the entire array. In addition, since all array variables are defined in sequential memory addresses, a machine language program can compute the variable

Single Dimension Array Memory Format
Figure 2



Relative Address to first element	Description
4	Start of second element of a single precision array or third element of an integer array.
2	Start of the second element of an integer array
Ø	First character of array data (first element string length in the case of a string).
-1	Number of elements in Array (MSB)
-2	Number of elements in Array (LSB)
-3	Number of dimmensions (1 for a one dimension array)
-4	Length of entire array data (MSB)
-5	Length of entire array data (LSB)
The following varies depending on the variable name.	Remaining characters in variable name
	Length of remaining variable name characters
	Second character of variable name
	First character of variable name
	Variable type (2-Integer, 3-string, 4-single precision number, 5-Double precision number)

descriptor address for any element in the array by using the formula:

Variable Descriptor=Variable descriptor of first element+ Data length*element number

where data length is '2' for integers, '3' for strings, '4' for single precision numbers, and '8' for double precision numbers. In Figure 2, the format of the first few bytes of an array is shown.

In many cases, a machine language subroutine will need to transfer information back to a BASIC program. If the information that needs to be transferred is numeric, the machine language program can easily change the value in any one of the numeric parameters passed. For example, if you wished to load the value in AX into the last integer variable in the parameter list, the following program could be used.

```
MOV    BP,SP          ;Make BP base of stack
MOV    BX,[BP+4]      ;Get last variable
                       descriptor
MOV    [BX],AX        ;Transfer AX into last
                       variable
```

While numeric variables are easy to modify, string variables are unfortunately a bit more difficult. If the data that is to be modified does not affect the length of the string, the data from the machine language subroutine can overwrite the data that is already in the string. If the length of the data needs to be changed, you must make sure that the string variable is in BASIC's string accumulator before modifying it. To do this, add a null string to the string that is to be modified in BASIC (example, A$=A$+"") just before CALLing the machine language subroutine. With the data in BASIC's string accumulator, you do not have to worry about expanding a string on top of another variable.

Once the machine language subroutine has completed its task, it must return to BASIC using a far RETurn with additions to the stack. The additions to the stack should be equal to two times the number of parameters passed. For example, a proper return from a subroutine with five parameters would be:

```
RET 1Ø
```

By adding to the stack after the return, all of the variable descriptors are forced off the stack. This keeps the stack balanced for further operations in BASIC.

Installing ML Programs in BASIC

When writing machine language subroutines for BASIC, a few precautionary steps need to be taken. First, it is a very good idea that the machine language subroutine be 100 percent relocatable (capable of executing anywhere in memory). Since BASIC and your machine language subroutine can be loaded almost anywhere in memory under MS-DOS, your routine must not assume any absolute memory addresses. On the 8088/186 this is quite simple since most instructions use relative addressing (relative to the current Instruction Pointer). The only type of commands you have to worry about are direct memory loads. If the machine language subroutine addresses memory inside itself for data storage purposes, the program is not 100 percent relocatable. If a machine language program needs to store data, it is usually best to use one of BASIC's variables or the stack as a temporary storage area.

Another item that you must remember with machine language subroutines for BASIC is to *make sure* that a far RETurn is used to return back to BASIC. Since the 8088/186 instruction name for a far and normal RETurn are identical (RET), the assembler might not choose the right type of RETurn. If you are using MS-Assembler, the easiest way of telling the assembler that you want a far RETurn is to define the entire machine language subroutine as a far PROCedure. Then, the assembler knows that the subroutine is to be accessed by a far CALL and will assemble the subroutine accordingly.

If a machine language subroutine conforms to the above two requirements, it should run fine under GW-BASIC. The only problem now is getting the machine language subroutine from a source file to your BASIC program. The first step, of course, is to assemble and link the machine language source file into a machine language .EXE file.

Next, find the length of the actual machine language subroutine from the output listing from the assembler. The length of the procedure should be listed on the last page of the output listing.

Once the length of the routine is found, the address where the program will load can be computed. If the subroutine is greater than 256 bytes long, the program will need to reside in the upper regions of BASIC's data memory. On a 256K+ machine, the address can be calculated by subtracting 65520 from the length of the program. The resulting number determines the load and start address of the machine language subroutine. To prevent BASIC from overwriting the machine language subroutine in upper memory with variables, BASIC needs to be told where its upper memory boundary should be. To do this, use the /M option when entering BASIC to set the high memory value to an address below the subroutine.

If the machine language subroutine is less than 256 bytes long, it can be loaded into a string. With this method, you don't have to worry about where the machine language program should be loaded. In addition, since BASIC will never destroy the contents of a string, your program is safe from being modified by the interpreter.

To find the address of a machine language subroutine after it has been loaded into a string, perform the following operation:

```
X=VARPTR(string name):subroutine address
=PEEK(X+1)+PEEK(X+2)*256
```

If you intend to load the machine language subroutine into a string, a few precautions need to be taken. First, make sure that the variable containing the machine language subroutine is the first to be defined by BASIC. Failure to do so could result in the subroutine moving all throughout RAM. In addition, make sure that a FRE(variable) function is executed before determining the address of the string containing the machine language subroutine. When the FRE( ) is performed, BASIC deallocates any unused string space. In doing this, all of BASIC's variables are moved around in memory to fill the freed space. Since you want to make sure your machine language subroutine does not move around after you find its address, it is a good idea to force BASIC to perform all its moving before you find the string's address.

With either method of storing machine language subroutines, the subroutine has to be transferred from a .EXE file into BASIC's data memory. One way of doing this is with BASIC's BLOAD instruction. This instruction allows a binary file to be loaded off disk into BASIC's data segment. The only problem with BLOAD is that it requires that the first 512 bytes of the .EXE file be chopped off before the file can be loaded into memory.

The other option (the option I prefer) is to convert the .EXE file into DATA statements which can be used by BASIC to load the subroutine into memory. This conversion process can be performed by the BASIC program in Program 1. To use this program, enter the name of the source .EXE file (no extention), the destination .BAS file (no extention) that will hold the data statments, and the starting line number. After a few seconds of compiling, the .BAS program is generated and saved on disk. The resulting .BAS program starts with a REM with the program name and machine langauge subroutine length followed by the machine language program converted into data statements. The data in these DATA statements can be read by BASIC's READ statement and either poked into high memory or stored in a string. If you are interested in how the .EXE to DATA statement converter operates, Table 1 gives a complete line by line description of the program.

Subroutines

Well, since the name of this series is "Subroutine City" I guess I had better describe a few subroutines.

One subroutine that I always seem to need in any program I write converts a string of text to uppercase. This is especially useful when you need to search through a list of data that has mixed upper- and lowercase characters. The only problem is that under normal BASIC, string uppercase conversions require quite a bit of time. To alleviate this problem, I wrote a machine language subroutine to perform the conversion in a matter of microseconds.

Program 2 contains the BASIC code for a machine language uppercase converter subroutine. This subroutine requires that one string parameter be passed to the routine on entry. On return from the subroutine, the lowercase text in the string is converted to uppercase.

Line 10 of Program 2 predefines the variable UC$; the string that holds the machine language subroutine. Line 20 loads the machine language subroutine into UC$ and the subroutine address in UC. After this point, the machine language subroutine is ready to use by your BASIC program. The demo program in Line 30 inputs a character string from the keyboard, converts it to uppercase with the CALL UC(A$), and prints the subroutine output on the screen.

Another function that is badly needed on the Model 2000 GW-BASIC allows the current directory to be changed. Under standard Model 2000 GW-BASIC, only the files on the currently selected directory can be used. This is a real problem when all the system languages and utilities are bunched under one directory and all the application programs and data are on other directories.

To solve this problem, Program 3 contains a short machine language subroutine that taps BASIC into MS-DOS' subdirectory change routine. This subroutine requires that the first parameter passed to the subroutine be a string containing the name of the new subdirectory followed by a null (CHR$(0)). The second parameter passes MS-DOS's error status after the operation back to BASIC. If the subroutine returns with a 3, the subdirectory selected does not exist on the current directory. If the subroutine returns with a 0, the directory was changed successfuly.

Line 10 of Program 3 defines and preallocates all the variables used by the machine language subroutine. Line 20 loads the machine language program into P$ and the subroutine address into AD. Line 30 contains a sample program that merely asks for a subdirectory, changes the subdirectory with the machine language subroutine, and prints the files that are on the new directory on the screen.

Both of the above machine language programs can be easily used in a single BASIC program. If you intend to do this, *make sure you define all of the string variables first*! Also make sure not to omit the FRE( ) instruction that precedes the subroutine address calculation logic.

Program 4 and Program 5 contain the MS-Assembler sources for the uppercase converter and directory changer.

If you are interested in writing your own machine language subroutines for BASIC, I suggest that you study these source programs carefully.

Conclusion

Now that I have described how to use machine language subroutines in BASIC, I plan to use them more in each month's installment of "Subroutine City."

In next month's installment, I will be describing an entire set of BASIC subroutines for accessing 95 percent of the functions available in MS-DOS. Every function from changing the current default drive to loading and executing .EXE files from BASIC will be discussed. As a matter of fact, after next month's installment, you will have enough extensions to BASIC to write your own DOS command processor in BASIC!

```
1 ' **** .EXE to DATA converter
2 ' ****  By Robert D. Covington
20 CLS:PRINT"EXE to BASIC DATA converter":PRINT"by Robert D. Covington":PRINT
30 LINE INPUT"Source Filename (.EXE):";S$:IF S$="" THEN 30
40 LINE INPUT"Destination Filename (.BAS):";D$:IF D$="" THEN 40
50 INPUT"At what BASIC line number should the data begin";DL:IF DL<1 OR DL>65000
! THEN PRINT"Line Number out of Range!!":GOTO 50
80 PRINT:PRINT"Converting Program";
100 OPEN"R",1,S$+".EXE",1:FIELD 1,1 AS X$:OPEN"O",2,D$+".BAS"
110 GET 1,3:L=ASC(X$):GET 1,4:L=L+ASC(X$)*256:IF L=0 THEN PRINT"No program!!":GO
TO 900
200 PRINT #2,DL;"' Program:";D$,"Length:";L;"bytes"
210 DL=DL+1:T$=STR$(DL)+"DATA "
300 FOR X=513 TO 513+L-1:ND=0:GET 1,X:T$=T$+MID$(STR$(ASC(X$)),2)+",":IF LEN(T$)
>230 THEN PRINT #2,LEFT$(T$,LEN(T$)-1):DL=DL+1:T$=STR$(DL)+" DATA ":ND=-1
310 PRINT".";:NEXT:IF NOT ND THEN PRINT #2,LEFT$(T$,LEN(T$)-1)
900 CLOSE:PRINT:PRINT"Conversion Complete":PRINT"Program size:";L;"bytes":END
65000 SAVE"EXE2DAT.BAS"


 1 ' **** Uppercase String converter
 2 ' **** By Robert D. Covington
 10 UC$=""
 20 FOR X=1 TO 38:READ A:UC$=UC$+CHR$(A):NEXT:X=FRE(UC$):V=VARPTR(UC$):UC=PEEK(V+
 1)+PEEK(V+2)*256
 30 INPUT"Enter String to convert to upper case";A$:CALL UC(A$):PRINT"Uppercase T
 ext=";A$:GOTO 30
 1000 ' Program:UC           Length: 38 bytes
 1001 DATA 139,236,139,94,4,181,0,138,15,10,201,116,22,139,111,1,138,70,0,60,97,1
 24,9,60,122,127,5,36,223,136,70,0,69,226,237,202,2,0


 1 '  **** Sub Directory Changer
 2 '  **** By Robert D. Covington
 10 DEFINT R:P$="":
 20 FOR X=1 TO 24:READ A:P$=P$+CHR$(A):NEXT:X=FRE(P$):V=VARPTR(P$):AD=PEEK(V+1)+P
```

Submitting Material to PCM

Contributions to PCM are welcome from everyone. We like to run a variety of programs which will be useful/helpful/fun for other Tandy Portable and MS-DOS computer owners. We now support the Model 100, the Tandy 200, and the Tandy models 2000, 1200 and 1000.

Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs. All programs should be supported by some editorial commentary explaining how the program works. Generally, we're much more interested in how your submission works and runs than how you developed it. Programs should be learning experiences.

Pay for submissions is based on a number of criteria. The rate of remuneration will be established and agreed upon prior to publication.

For the benefit of those who wish more detailed information on making submissions, please send a SASE to: Submissions Editor, PCM, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit programs or articles currently submitted to another publication.

If you feel qualified to review software and/or hardware products for computers covered in PCM, send us your name, address and phone number; we will send you a questionnaire form and a copy of our reviewer guidelines.

```
EEK(V+2)*256
30 INPUT"New Subdirectory";A$:A$=A$+CHR$(0):CALL AD(A$,R):PRINT"RETURN STATUS=";
R:PRINT:PRINT"Files on new Subdirectory:":FILES:PRINT:GOTO 30
2000 ' Program:SUBDIR      Length: 24 bytes
2001 DATA 139,236,139,94,6,139,87,1,180,59,205,33,139,94,4,114,2,51,192,137,7,20
2,4,0
```

```
         PAGE     6Ø,8Ø
         TITLE    LC to UC String converter for BASIC
CODE     SEGMENT
         ASSUME   CS:CODE,DS:CODE,ES:CODE
CONVERT  PROC     FAR                ;Define procedure
START:
         MOV      BP,SP              ;Make BP base pointer of stack
         MOV      BX,[BP+4]          ;Get string descriptor for 1st param.
         MOV      CH,Ø               ;Clear MSB
         MOV      CL,[BX]            ;Put String length in CX
         OR       CL,CL              ;Check for CL=Ø
         JZ       BYE                ;Exit if string empty
         MOV      BP,[BX+1]          ;Put String address in BP
CLOOP:
         MOV      AL,[BP]            ;Get character in string
         CMP      AL,'a'             ;Check lower character limit
         JL       SKIP               ;Skip if less than 'a'
         CMP      AL,'z'             ;Check for upper character limit
         JG       SKIP               ;Jump if greater than 'z'
         AND      AL,223             ;Convert character to uppercase
         MOV      [BP],AL            ;Resave converted character
SKIP:
         INC      BP                 ;Increment string pointer
         LOOP     CLOOP              ;Loop till entire string converted
BYE:
         RET      2                  ;Return to BASIC
CONVERT  ENDP
CODE     ENDS
         END      START
```

```
         PAGE     6Ø,8Ø
         TITLE    Subdirectory changer for BASIC
CODE     SEGMENT
         ASSUME   CS:CODE,DS:CODE,ES:CODE
DIR      PROC     FAR                ;Define procedure
START:
         MOV      BP,SP              ;Make BP base pointer of stack
         MOV      BX,[BP+6]          ;Get string descriptor for 1st param
         MOV      DX,[BX+1]          ;Point DX to string
         MOV      AH,3BH             ;Set Change directory routine
         INT      21H                ;Change directory
         MOV      BX,[BP+4]          ;Get address for return integer
         JC       ERROR              ;Jump if error
         XOR      AX,AX              ;Clear error status (AX=Ø)
ERROR:
         MOV      [BX],AX            ;Save return status
         RET      4                  ;Return to BASIC
DIR      ENDP
CODE     ENDS
         END      START
```

What sort of sort is best for your sorting needs? PCM Contributing Editor Bill Barden is just the sort of guy who can help answer that question . . .

Adventures of a Different Sort

By William Barden, Jr.
PCM Contributing Editor

It was yet another meeting of the Orange County Tandy User's Group, a conglomeration of wild-eyed computer hackers, old-time Tandy aficionados who had switched from Model IIIs and 4s to the 1000 and 1200HD, and befuddled beginners who had been duped by Tandy Computer Center salesmen into thinking that running a 2000 was roughly equivalent to operating a Cuisinart. It was question and answer time and a new Tandy 1000 owner had the floor:

"I have a mail list of about 500 names that I have to sort by ZIP. How do I do it?"

"Use the MS-DOS SORT utility," someone advised.

"But the names aren't organized in fields! The SORT won't work," the 1000 owner lamented.

"Write your own sort in BASIC," Hal the Hacker shouted. "It's easy — just use a Quicksort."

"Quicksort! Why not a bubble sort?," someone else contributed.

"Naw, a bubble sort's too slow," another member said. "Try the sort that was in PCM Magazine a few months ago."

I smiled to myself and thought how history repeats itself. The whole scene was a replay of the early days of computing. Believe it or not, *volumes* have been written and thousands of articles have been published in scholarly computing journals about sorting. It seemed for a while everyone had his favorite sort. In this column we'll take a look at some of the mystery about sorting to see whether it's feasible to actually do your own sort in BASIC, and if so, which sort to use. You'll probably be surprised at some of the results, especially when the results are compared to the SORT utility in MS-DOS.

(William Barden, Jr., is a master communicator in a field in which he is one of the few recognized experts — microcomputers. A prolific author of more than 27 books and handbooks on computers and computer programming, Bill also has authored several instructional software projects for Tandy/Radio Shack.)

What is Sorting?

The mail list is a good example of why sorting is necessary — bulk mail has to be separated by ZIP code. A convenient way to do this is to have a computer print out address labels in ZIP code order. Of course, many other things also need to be in some kind of order — telephone directories, test scores, income tax returns, and encyclopedias, to name a few. It's been estimated that 25 percent of all "computer time" is devoted to sorting — putting lists in some kind of order. Efficient sorting has been a major concern since the early days of computing, in the 1940s.

One of the standard reference works for sorting is a book called *Sorting and Searching* by Donald E. Knuth (Addison Wesley). Knuth is a professor of computer science at Stanford who has written many other volumes in "The Art of Computer Programming" series. *Sorting and Searching* goes into a great deal of detail about these subjects and references papers published from the 1940s to the present day. Under sorting methods, Knuth lists "Binary insertion sort, Bitonic sort, Bubble sort, Cocktail shaker sort, Comparison counting sort, Distribution counting sort, Heapsort . . . " Anyway, you get the idea — there's a whole lotta sortin' going on! We'll look at some of the less esoteric sorts from Knuth here — they're fairly easy to understand and can have a real impact on your applications.

ASCII Codes, Strings, and Upper- and Lowercase

Sorting *numeric* data is usually not too much of a problem in small computers. Numeric data is more compact than text data and can be handled with less overhead. All of the techniques discussed here apply to numeric data as well, but we'll stick with the more troublesome case of text data for our examples.

Text data has a numeric value based on the ASCII characters in the data. ASCII values for uppercase A through Z are 65 through 90, for 0 through 9 are 48 through 57, and for lowcase a through z are 97 through 122 (all in decimal). Special characters, such as #, %, and < also have ASCII values. The order of displayable ASCII codes is:

> (space)!"#$%&'( )*+,-./0123456789:;<=>?@ABC DEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefgh ijklmnopqrstuvwxyz{¦}~

Here's a sort based on ASCII values:

APPLE II
APPLE III
Acme Computer II+4
Acme Computer II-A
Cray I
IBM PC
IBM PC*jr*
TRS-80 Model I
TRS-80 Model III
apple //c

There are several interesting things that happen in an "ascending" (A-Z) sort of this type. Note that the string "APPLE III" is ordered after "APPLE II." This is because the "APPLE III" text string has a greater "weight" than the "APPLE II" as it's longer — it's like the values 10034 and 10034.7. Note also that special characters like commas and periods aren't ignored in the sort — "Acme Computer II+4" comes before "Acme Computer II-A," for example, because of the + character. Finally, and this causes a lot of trouble, lowercase characters sort *after* uppercase characters! The apple //c entry is at the end of the list, out of alphabetical order. Sorts have to build in some kind of logic to compare upper- and lowercase characters on an equal basis, unless a standard format is used.

The MS-DOS SORT Utility

The Tandy 1000, 1200HD, and 2000 have a "built-in" sort called SORT. It's a machine-language sort that generally operates on a disk file, although you could, for example, use it to sort data entered from a keyboard. The results of the sort generally go to another disk file or the display. Here are some formats of SORT:

A>SORT <names.dat	sorts file NAMES.DAT to display
A>SORT <names.dat >names1.dat	sorts file NAMES.DAT to file NAMES1.DAT
A>SORT /R <names.dat	sorts file NAMES.DAT to display but in reverse (Z-A)
A>SORT /+5 <names.dat	sorts file NAMES.DAT to display starting with the fifth character

The last option of SORT needs some explanation. A lot of computer lists are of two different types — "fixed" field and "variable-length" field. A fixed field list looks like this:

```
12345678901234567890123456789012345678901 . . .
Barden       William     28122 Orsola    M . . .
```

In this format, first names always start at the 14th character position, street addresses always start at the 26th character position, and so forth. Contrast this type of list entry with:

```
  Barden/William/28122 Orsola/ . . .
```

where the "fields" of the entry are of variable length and are separated by a slash mark.

The SORT utility is designed to be used with the first type of format, where each field — last name, first name, street, city, state, ZIP — have a fixed number of characters. Doing a

```
  A>SORT /+26 <names.dat
```

in the example above would sort the list by street address, ignoring everything that precedes the 26th character.

Just, how fast is the SORT utility? To test that, I generated a list of 200 mailing list names using a random list generator program. Typical entries in the list look like this:

```
Mrs. Margaret Fletcher  753  Ave of the Stars
     Dayton       WI    53182
Mr.  William White      1234  Le Conte
     St. Louis    OR    97204
Mr.  Timothy Black       888  Le Conte
     Portland     CA    94101
Mr.  Ronald Hunter     10000  Pico Blvd.
     Miami        TX    76102
```

SORT is an "in memory" sort, so I estimated the time at which disk activity stopped and then timed the sort up to the point at which screen display began. The SORT on my Tandy 1000 did the task in about 12 seconds. We'll use this time as a benchmark against which to compare the BASIC sorts discussed here. Can we actually implement a BASIC sort which is as fast as the SORT utility? Read on . . .

The Infamous Bubble Sort

Probably the most notorious sort in BASIC is the "bubble sort", pictured in Figure 1. The bubble sort works like this: There's originally an unsorted list of items (in this figure numeric). Starting from the top of the list, a *pair* of items is compared. If the bottom item is of smaller value than the top item, the items are *swapped.* If the bottom item is of larger or equal value to the top item, nothing is done. The next two items (the prior bottom item and the next item) are then compared in the same fashion and a swap is made if necessary. This process continues until the last items in the list have been compared. At the end of the comparisons either the list was *ordered* or one or more swaps has occurred. If one or more swaps occurred, the entire process repeats, starting from the top again. The process is repeated until no swaps occurred during a pass, indicating the entire list is in order.

The name bubble sort is given to the scheme because "lighter" entries "bubble" to the top of the list.

The bubble sort is popular because it's simple and elegant. Another feature that makes it handy to use is that it only uses the original data area — no new area has to be set aside to hold entries that are being processed. However, the bubble sort has one glaring fault — it's horrendously slow compared to other sorts we'll be discussing. The longer the list, the slower it becomes (as for all sorts discussed here), as shown in Figure 2, so you might want to use it for sorting a few dozen entries, but certainly not hundreds.

Listing 1 shows the bubble sort implemented for our mailing list simulation. The actual sort is only six BASIC lines. The remainder of the program reads in the 200 mailing list entries into array AR$.

Figure 2. Typical Sort Times Versus Number of Entries on Tandy 1000/1200

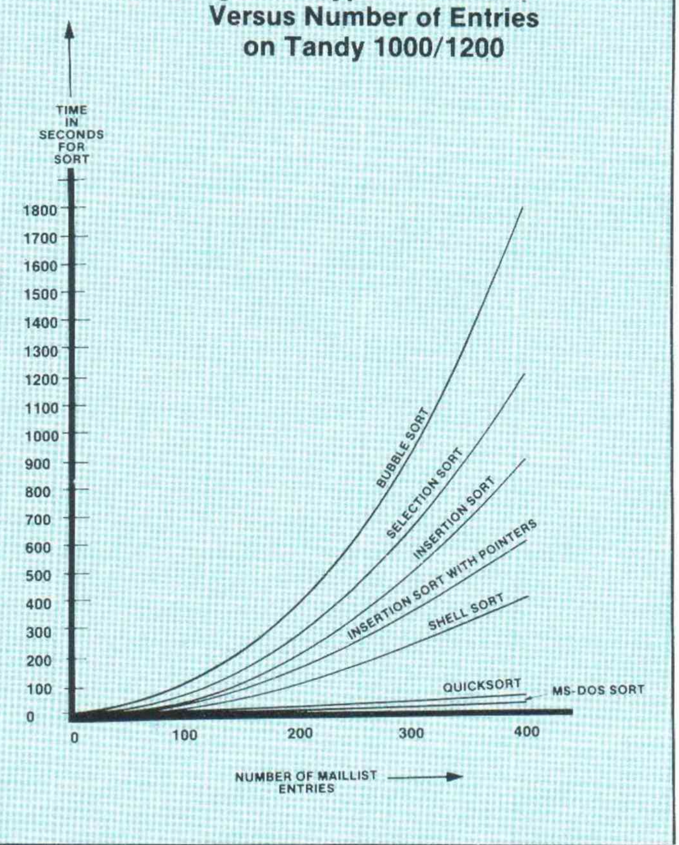
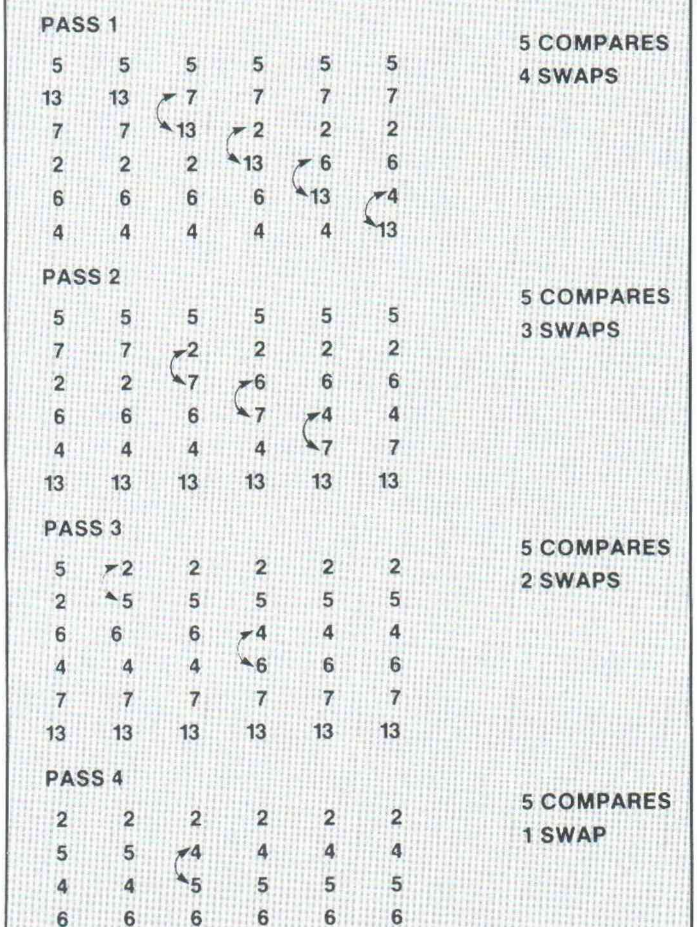


Figure 1. Bubble Sort Operation



PASS 1

5	5	5	5	5	5
13	13	7	7	7	7
7	7	13	2	2	2
2	2	2	13	6	6
6	6	6	6	13	4
4	4	4	4	4	13

5 COMPARES
4 SWAPS

PASS 2

5	5	5	5	5	5
7	7	2	2	2	2
2	2	7	6	6	6
6	6	6	7	4	4
4	4	4	4	7	7
13	13	13	13	13	13

5 COMPARES
3 SWAPS

PASS 3

5	2	2	2	2	2
2	5	5	5	5	5
6	6	6	4	4	4
4	4	4	6	6	6
7	7	7	7	7	7
13	13	13	13	13	13

5 COMPARES
2 SWAPS

PASS 4

2	2	2	2	2	2
5	5	4	4	4	4
4	4	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
13	13	13	13	13	13

5 COMPARES
1 SWAP

The program scans array AR$ from entry 0 through entry 199. The FOR loop is for one less than the last entry because pairs are being considered. If any swap of pairs is necessary, a SWAP is done and variable CHANGE is set to a 1. At the end of the loop, the process is repeated if CHANGE is a 1. If CHANGE is a 0, no swap was made and the list is sorted. A period is displayed for every swap, and a new line is displayed for every "pass" through the loop. It's fun to watch the number of swaps (periods) diminish as the program is executed.

For the 200 entries in our sample list, the bubble sort of Listing 1 took 537 seconds, about 44 times longer than the SORT utility.

A Selection Sort

The selection sort works the same way you might sort a list by manual methods. If you wanted to alphabetize a telephone list, for example, you might scan the list looking for all 'A' entries. You'd then put these entries in a second list, starting at the top, and cross off the 'A' entries on the first list. You'd then look for the 'B' entries and do the same thing, repeating the process until all entries were crossed off. See Figure 3.

Listing 2 shows a BASIC version of the selection sort. Two arrays are used, as shown in Figure 3. Array AR$ holds the original mailing list values, read in with code identical to the bubble sort. Array AT$ will hold the sorted entries at the end of the sort. Two variables point to entries in the arrays. Variable I is a place marker for the sorted array. Entries are put in order from 0 through 199 as they are found. Variable J is used to "scan" the unsorted entries

Listing 1:

```
100 ' BUBBLE SORT
110 ' initial setup
120 DIM AR$(199)
130 OPEN "maillist" FOR INPUT AS #1
140 I=0
150 IF EOF(1) THEN 180
160 LINE INPUT#1, AR$(I)
170 I=I+1: GOTO 150
180 CLOSE: PRINT TIME$
190 ' actual sort
200 PRINT: CHANGE=0
210 FOR I=0 TO 198
220 IF AR$(I)>AR$(I+1) THEN SWAP AR$(I),AR$(I+1): CHANGE=1: PRINT ".";
230 NEXT I
240 IF CHANGE=1 THEN 200
250 PRINT TIME$
```

in array AR$ finding the entry of lowest value (smallest text string weight). After each pass through the 'J' loop, variable SMALLEST holds the index, 0 through 199, of the smallest entry in array AR$.

Listing 2:

```
100 ' SELECTION SORT
110 ' initial setup
120 DIM AR$(199), AT$(199)
130 OPEN "maillist" FOR INPUT AS #1
140 I=0
150 IF EOF(1) THEN 180
160 LINE INPUT#1, AR$(I)
170 I=I+1: GOTO 150
180 CLOSE: PRINT TIME$: SMALLEST=0
190 'actual sort
200 FOR I=0 TO 199
210 PRINT ".";
220 FOR J=0 TO 199
230 IF AR$(J)<AR$(SMALLEST) THEN SMALLEST=J
240 NEXT J
250 AT$(I)=AR$(SMALLEST): AR$(SMALLEST)=CHR$(255)
260 NEXT I
270 PRINT TIME$
```

After each pass, the smallest entry from array AR$ is transferred to the next position of array AT$. The entry from array AR$ is then "crossed off" by setting it to the one character string CHR$(255), a value that will always be higher than any text string and therefore never found. As each pass is made, a period is displayed on the screen.

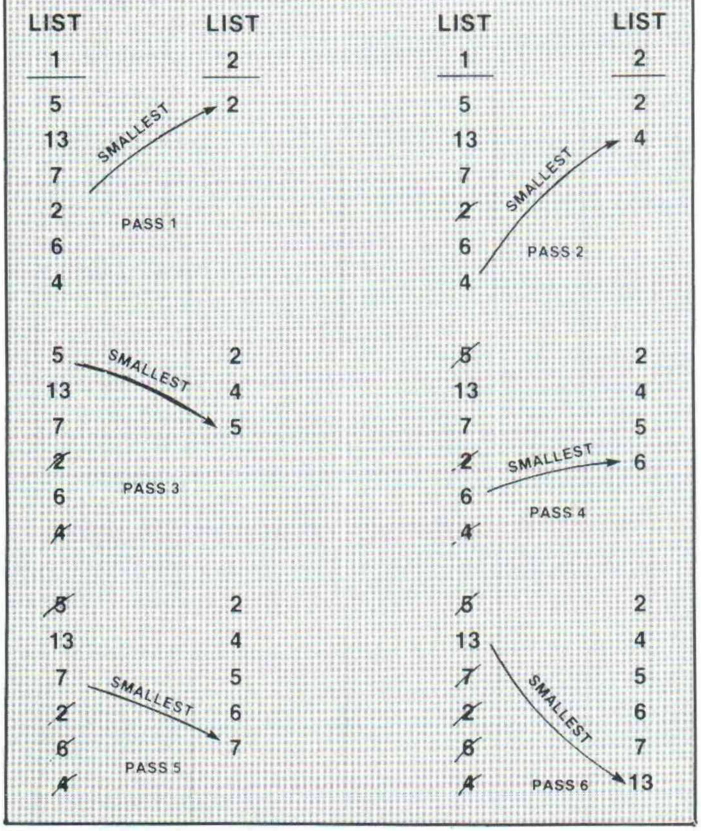
The selection sort does much better than the bubble sort at about 363 seconds. The disadvantage of the selection sort is that it uses two storage areas. This space problem is alleviated somewhat, however, as the one-character "cross off" string releases most of the space required by the deleted entry to the common area used by BASIC.

It might be a good time to mention, by the way, that this common area is used by BASIC as a string storage area. Periodically, BASIC "cleans up" any garbage strings in the area to make room for new strings. This is most noticeable when you are using large areas of storage and do a lot of string processing. In many cases it will result in long delays where it appears as if nothing is happening. To be fair about it, these times should be subtracted from sort times. On the other hand, we're discussing practical situations here, in specific systems, so possibly the "garbage collection" times should be considered. We've chosen the former course here.

A Straight Insertion Sort

The insertion sort also uses a second storage area. The original list entries are taken one at a time starting from the top of the list, in their unordered state. This is an advantage, by the way, because it means that the data does not have to be "in memory," but can be read in on the

Figure 3. Selection Sort Operation



fly. As each entry is taken from the list, it is inserted in the second storage area at the proper point, as shown in Figure 4.

To insert the new entry means that the list of items in the second storage area must be "moved down" to make room for the new entry. Unfortunately, this requires a lot of time in BASIC (or any language). The program in Listing 3 shows the implementation. There is a great deal of "overhead" in moving data here, even though the 287 second sort time is better than that of the selection sort.

The program reads in the mail list data file into array AR$ as in the prior examples. It then goes down array AR$ from top to bottom picking up each entry by the I variable index. After each entry is picked up, the second storage area is scanned from the top (J=0) to the LAST entry to find the insertion point, the point at which the entry should "squeeze in." When the point is found, the prior entries from J+1 to LAST are moved down one position. This operation must be in reverse, however, to avoid *clobbering* the entries as they are moved. After the move, the AR$ entry is put into the AT$ array at location AT$(J). The AR$ entry is then deleted by setting it to null string ("") to release the string area.

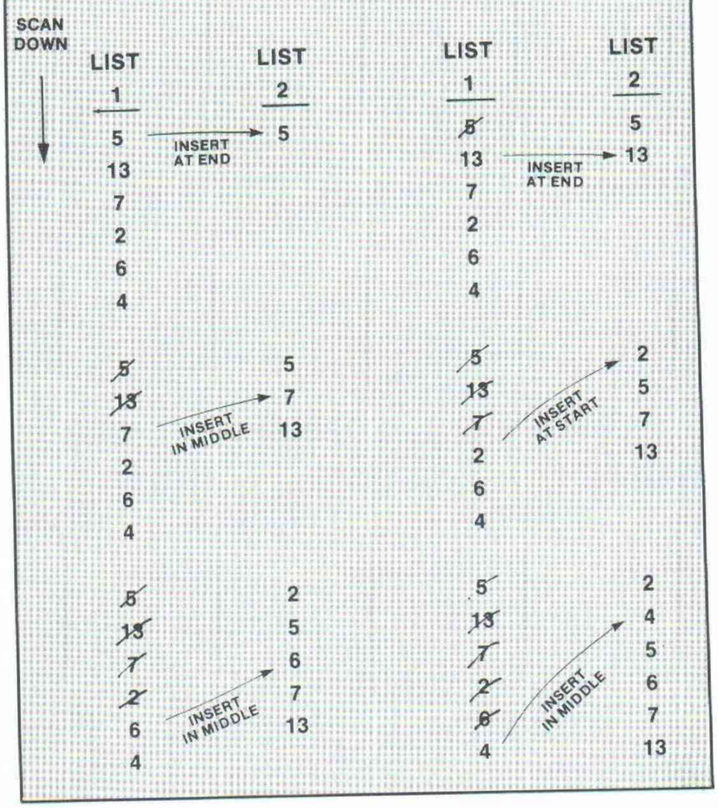
A special case occurs when the last entry is less than the entry to be inserted. In this case, the entry is appended to the list and no move is done.

An Improved Insertion Sort

The insertion sort is a good idea, but burdened down with time-consuming data movement. The program in Listing 4 reduces the data movement by using "pointers" associated with each entry, as shown in Figure 5. At any time, the pointers form a thread defining the order of the list, as shown in the figure. The list is scanned in order by using pointers as "indices" to the arrangement of the string array. A new entry is inserted by moving the pointers down as in the prior insertion sort. As the pointers are "integer" values, the move takes much less time than moving large string blocks.

The program listing is shown in Listing 4. The data file is read in as in the prior examples. The AR$ array is then scanned from top to bottom, using the I variable. Each pass picks up the current entry from AR$. This entry is then compared with the current entries in an imaginary second list, the entries being defined by the pointer list in array PNT%. When the insertion point in PNT% is found, the pointers in PNT% are moved down and the current pointer value is inserted. The special "append" case is handled similarly to the Listing 3 case.

Figure 4. Insertion Sort Operation



Listing 3:

```
100 ' STRAIGHT INSERTION SORT
110 ' initial setup
120 DIM AR$(199), AT$(199)
130 OPEN "maillist" FOR INPUT AS #1
140 I=0
150 IF EOF(1) THEN 180
160 LINE INPUT#1, AR$(I)
170 I=I+1: GOTO 150
180 CLOSE: PRINT TIME$: LAST=0: AT$(0)=CHR$(255)
190 ' actual sort
200 FOR I=0 TO 199
210 PRINT ".";
220 FOR J=0 TO LAST-1
230 IF AR$(I)>AT$(J) THEN 250
240 FOR K=LAST TO J+1 STEP -1: AT$(K)=AT$(K-1): NEXT K: AT$(J)=AR$(I): AR$(I)=""
: GOTO 270
250 NEXT J
260 AT$(LAST)=AR$(I)
270 LAST=LAST+1: NEXT I
280 PRINT TIME$
```

At the end of the scan of AR$, the threaded list of PNT% defines the order of the AR$ entries. To put this on a par with the other sorts, a new list with actual entries is made in AT$ by transferring the AR$ entries, although this need not necessarily be done. This version of the straight insertion sort executes in 196 seconds, quite an improvement over the program in Listing 3, and the leader to this point — almost three times faster than the bubble sort, but still 16 times slower than the MS-DOS SORT!

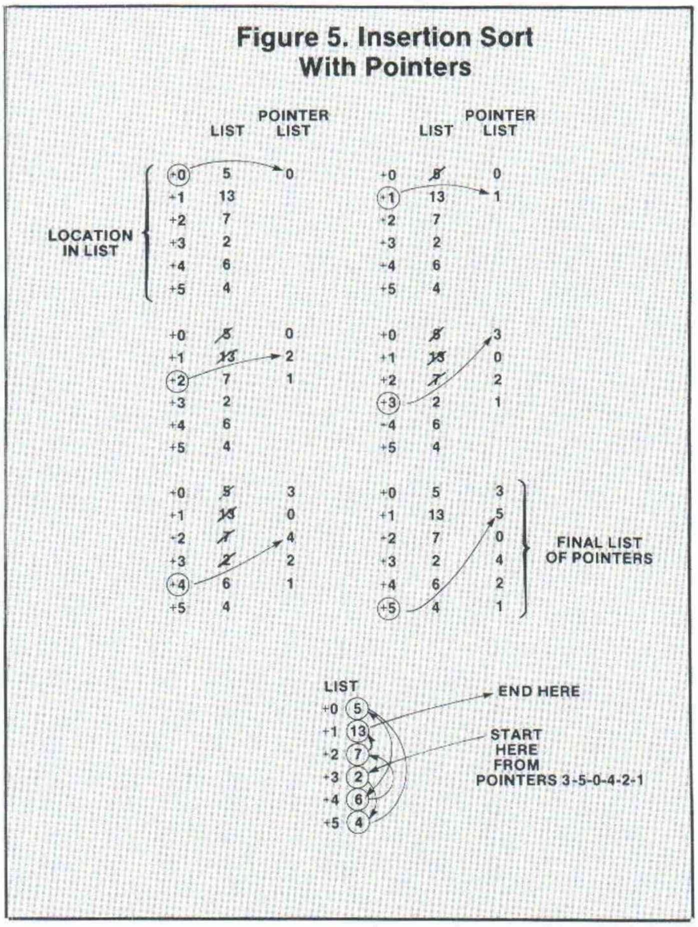


Figure 5. Insertion Sort With Pointers

The Shell Sort

At this point while writing the column, I'll admit I became obsessed with finding a faster BASIC sort that would compare with the SORT utility speed. There are many sorts in Knuth, and most are rather elaborate to achieve a great speed. I wanted something simple, elegant and fast. The Shell sort seemed a likely candidate.

The Shell sort (named for Donald L. Shell who extolled its virtues in 1959) is somewhat similar to the bubble sort. The problem with the bubble sort is that it takes a long time for an entry at the bottom of the list to bubble up to the top and many, many swaps. Shell increased the efficiency of the bubble sort by shortening the paths somewhat, making it possible for low-valued entries at the bottom to bubble up in a fraction of the swaps. To do this, he used "diminishing increments" for paths to the top, as shown in Figure 6.

The first increment is large, typically about three-tenths of the size of the list. Lower value entries move rapidly to the top as a path is established for each "position" within the increment width.

Successive increments are smaller, typically decreasing to one-half the size of the preceding increment, as shown in the figure. The last increment is an increment of one. Each of the successive increments is, in effect, a minor bubble sort. The last increment of one ensures that no entries are left unordered.

The listing for the Shell sort is shown in Listing 5. It is surprisingly short, after the usual read of data from the mail list file. For the 200-entry mail list file here, the increment starts at 65. Odd increments are used — 65, 33, 17, 9, 5, 3, and finally 1 (increment values are somewhat arbitrary). For each increment, the AR$ array is divided into a number of "widths" depending upon the increment size. The I variable moves across the width. For every increment, successive passes similar to the bubble sort are made until no changes have been made (CHANGE=0). When the increment is one and no change has occurred, the sort is over.

The Shell sort here takes 164 seconds, a good

Listing 4:

```
100 ' STRAIGHT INSERTION SORT WITH POINTERS
110 ' initial setup
120 DIM AR$(199), AT$(199), PNT%(199)
130 OPEN "maillist" FOR INPUT AS #1
140 I=0
150 IF EOF(1) THEN 180
160 LINE INPUT#1, AR$(I)
170 I=I+1: GOTO 150
180 CLOSE: PRINT TIME$: LAST=0: AT$(0)=CHR$(255): PNT%(0)=0
190 ' actual sort
200 FOR I=0 TO 199
210 PRINT ".";
220 FOR J=0 TO LAST-1
230 IF AR$(I)>AR$(PNT%(J)) THEN 250
240 FOR K=LAST TO J+1 STEP -1: PNT%(K)=PNT%(K-1): NEXT K: PNT%(J)=I: GOTO 270
250 NEXT J
260 PNT%(LAST)=I
270 LAST=LAST+1: NEXT I
280 FOR I=0 TO 199: AT$(I)=AR$(PNT%(I)): AR$(PNT%(I))="": NEXT I
290 PRINT TIME$
```

improvement over the modified insertion sort. I was getting closer to the ultimate goal of beating the SORT utility, but still wasn't there. However, I had an ace up my sleeve that I hadn't played yet. Do you know what it is?

Quick, Quicker, Quicksort

Looking through Knuth, I encountered Quicksort (C.A.R. Hoare, 1962). I had heard a lot about Quicksort in the past, but hadn't actually implemented it on a system, as it seemed somewhat esoteric. It *is* more difficult to understand than the other sorts, but is well worth the trouble. Quicksort is shown in Listing 6.

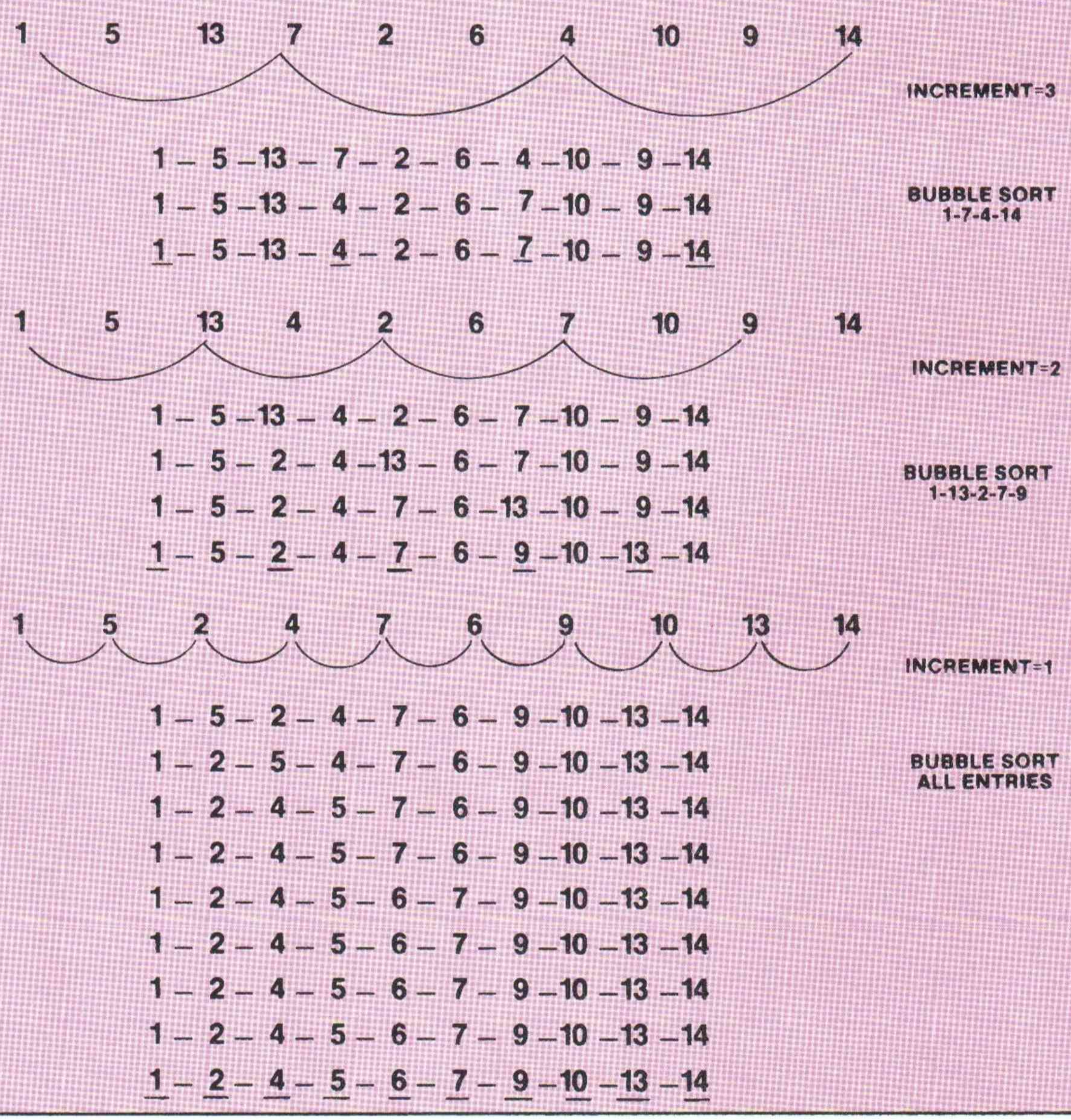
The algorithm (plan) is shown in Figure 7. It works like this: Starting with an arbitrary value (usually the first item in the list), divide the unordered list into two groups. The group on the left contains all items *less than* the value of the first item. The group on the right contains all of the items *greater than* the value of the first item. The first item is then moved between the two groups.

The locations of the two groups is then recorded. Next, the smaller of the two groups is "partitioned" in the same

Listing 5:

```
100 ' SHELL SORT
110 ' initial setup
120 DIM AR$(199)
130 OPEN "maillist" FOR INPUT AS #1
140 I=0
150 IF EOF(1) THEN 180
160 LINE INPUT#1, AR$(I)
170 I=I+1: GOTO 150
180 CLOSE: PRINT TIME$: INC=128
190 ' actual sort
200 INC=INT(INC/2) OR 1: PRINT INC
210 CHANGE=0: PRINT ".";
220 FOR I=0 TO INC-1
230 FOR J=0 TO INT(199/INC)-1
240 K=I+(J*INC): L=I+((J+1)*INC): IF L>199 THEN 260
250 IF AR$(K)>AR$(L) THEN SWAP AR$(K),AR$(L): CHANGE=1
260 NEXT J
270 NEXT I
280 IF CHANGE=1 THEN 210
290 IF INC<>1 THEN 200
300 PRINT TIME$
```

Figure 6. Shell Sort Operation



manner by using a new starting value. This leads to a smaller two groups. This process continues, each group being broken up into smaller groups until the group consists of only two entries in order. At any given time there might be several groups of varying sizes waiting to be processed. However, although individual groups are probably unordered, the groups themselves are all located in the proper position relative to the other groups. A minimum of swapping is done with Quicksort, and this reduces the overhead, making the sort very fast.

The Quicksort listing may be somewhat difficult to follow compared to the other sorts. There is only one array, with entries being swapped to other positions of the array. There are two "pointers" in the program — 'I' records the left position of the current group being processed, while 'J' records the right position of the current group being processed. These pointers move toward each other, to the right and left, respectively, as the group is processed. When the pointers "cross," the current group is ordered. The STACK array records the positions of other groups to be processed.

Typically, there may be one to six groups defined in the STACK. Smaller groups are processed first. When the STACK array is empty (NEXTV=0), all groups have been ordered, and the entire list is ordered.

Although a little more complicated, Quicksort executes in an astonishing 35 seconds for the 200-entry mail list test case — 15 times faster than the bubble sort, and five times faster than its closest competitor. Oh, and I might add, only 23 seconds slower than the MS-DOS SORT utility!

You're probably wondering what my hole card is . . . All of the sorts here have been timed in *interpreted* BASIC. Once a sort is checked out, it can be *compiled* and executed. The IBM BASIC Compiler runs on both my Tandy 1000 and 1200HD. When Quicksort was compiled and executed, it ran the 200-entry mail list sort in under one second, 12 times faster than the MS-DOS SORT! Obviously, the other sorts can be implemented in machine language, as well, but it's interesting to note that a BASIC sort can indeed be competitive with MS-DOS SORT.

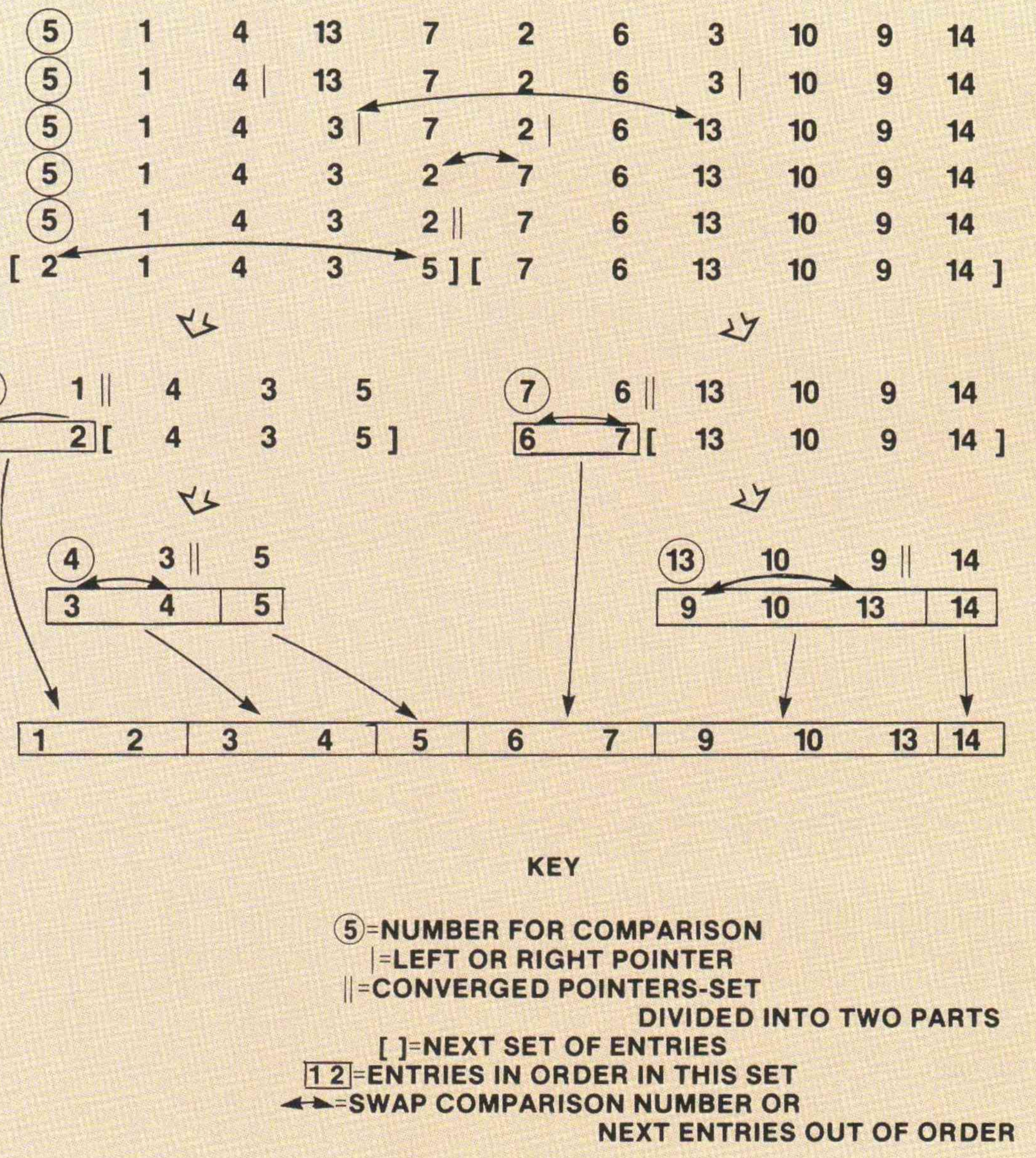
Sorting on Fields

Any of the sorts discussed here can be changed very easily to sort on a "field" start. Comparisons can be done

Listing 6:

```
100 ' QUICKSORT
110 ' initial setup
120 DEFINT B-Z
130 DIM AR$(199),STACK(30)
140 OPEN "maillist" FOR INPUT AS #1
150 I=0
160 IF EOF(1) THEN 190
170 LINE INPUT#1, AR$(I)
180 I=I+1: GOTO 160
190 CLOSE: PRINT TIME$
200 ' actual sort
210 NEXTV=2:STACK(0)=0:STACK(1)=199
220 IF NEXTV=0 THEN 420 ELSE THIS=STACK(NEXTV-2)
230 PRINT ".";
240 ' partition next i through j entries using ith entry as comparison value
250 I=STACK(NEXTV-2)+1: J=STACK(NEXTV-1): IF I>J THEN NEXTV=NEXTV-2: GOTO 220
260 IF AR$(I)>AR$(THIS) THEN 280
270 I=I+1: IF I>J THEN 330 ELSE 260
280 IF AR$(J)<AR$(THIS) THEN 310
290 J=J-1: IF I>J THEN 330 ELSE 280
300 ' swap to order two entries. continue 'til pointers cross
310 SWAP AR$(I),AR$(J): I=I+1: J=J-1: IF I>J THEN 330 ELSE GOTO 260
320 ' pointers have crossed. now exchange comparison value with ith entry
330 IF J<STACK(NEXTV-2) THEN J=STACK(NEXTV-2)
340 IF I>STACK(NEXTV-1) THEN I=STACK(NEXTV-1)
350 SWAP I,J: SWAP AR$(THIS),AR$(I)
360 ' now have 0, 1, or 2 segments. test and save in stack, largest first
370 K=STACK(NEXTV-2): L=STACK(NEXTV-1): NEXTV=NEXTV-2
380 IF I-K<=0 THEN IF L-J<=0 THEN 220 ELSE STACK(NEXTV)=J:STACK(NEXTV+1)=L: NEXT
V=NEXTV+2: GOTO 220
390 IF L-J<=0 THEN STACK(NEXTV)=K:STACK(NEXTV+1)=I-1: NEXTV=NEXTV+2: GOTO 220
400 IF I-K>L-J+1 THEN STACK(NEXTV)=K:STACK(NEXTV+1)=I-1: STACK(NEXTV+2)=J: STACK
(NEXTV+3)=L: NEXTV=NEXTV+4: GOTO 220
410 STACK(NEXTV)=J:STACK(NEXTV+1)=L: STACK(NEXTV+2)=K: STACK(NEXTV+3)=I-1: NEXTV
=NEXTV+4: GOTO 220
420 PRINT TIME$
```

Figure 7. Quicksort Example



using a RIGHT$ statement to use only the rightmost 'n' characters of each string. Similarly, the sense of the comparisons can be changed to do a "reverse" sort (Z to A). In addition, you can tailor the BASIC code for your own types of data; say, looking for delimiters (such as a slash between fields), sorting on more than one key (ZIP followed by name, for instance), or comparing upper- and lowercase.

The programs here are meant to be examples of typical implementations and not the most efficient design of some of the sorts. I'd be interested in hearing from readers about their sorting experiments and successes. Readers with sorted stories to tell (sorry) will be mentioned in dispatches in PCM Magazine. Write to me at P.O. Box 3568, Mission Viejo, CA 92692. I'll be waiting to hear from you about your 1.2 second "Heapsort." PCM

50
10
20
30
40
ALARM
WEST
GERMANY

A SCHEDL *Enhancement*

By Jay Halcomb

. . . adding graphics, an alarm and editing features

The Model 100 *SCHEDL* program is handy for keeping track of appointments, but it doesn't show you how your time is being used in a very graphic manner. I wrote Schedl.ba to enhance the *SCHEDL* program by doing just this — displaying a calendar of appointments and the daily schedule. Also, for convenience, the program enters and deletes appointments in the Note.do file, keeping the appointments sorted by date and hour. Finally, the program allows you to use the Model 100 as an alarm clock, "beeping" you at a preset time with any message you like.

The program is operated through the function keys. On first being run, the program presents you with a display of the current month's calendar with the dates on which you have appointments bracketed, and with the labels for the function keys: Calendar, Daily, Appointment, Delete, Alarm, Menu.

Pressing the function key under the label performs the appropriate action: Calendar prompts you for a month and year, then displays the calendar for that month with your appointments marked; Daily prompts you for the date, then displays the appointments you have on that date; Appointment prompts you for date, hour and details, then makes a record of the appointment in the note file; Delete prompts you for the date and hour, then deletes any appointment at that time from the note file; Alarm prompts you for the time you want to be "beeped," and for a message to display at that time, then sets the alarm; finally, Menu returns you to the Model 100 main menu.

Merely pressing ENTER when in the Calendar and Daily modes causes the program to assume the current dates. Pressing ENTER in the Appointment, Delete and Alarm modes returns you to the main selection function. Any time you wish to, you can copy the screen to the printer using the PRINT key on the Model 100.

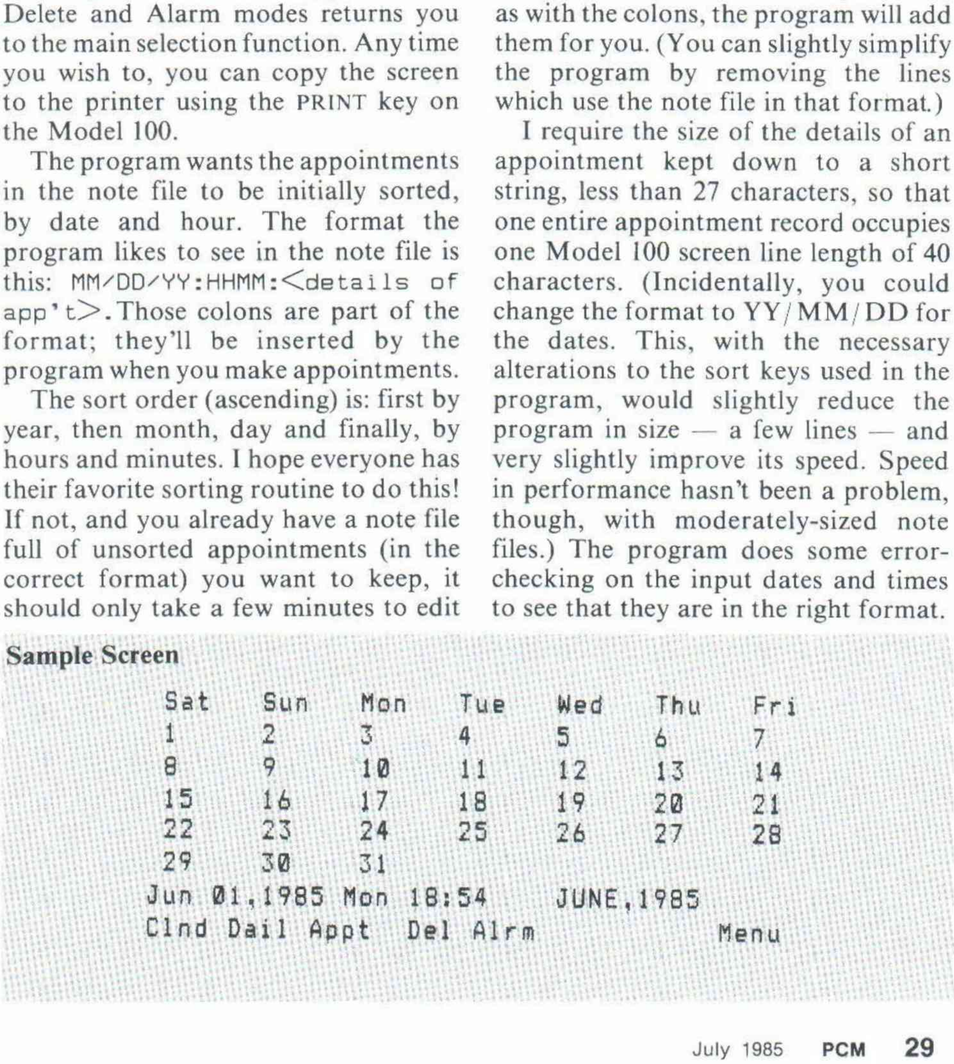
The program wants the appointments in the note file to be initially sorted, by date and hour. The format the program likes to see in the note file is this: MM/DD/YY:HHMM:<details of app't>. Those colons are part of the format; they'll be inserted by the program when you make appointments.

The sort order (ascending) is: first by year, then month, day and finally, by hours and minutes. I hope everyone has their favorite sorting routine to do this! If not, and you already have a note file full of unsorted appointments (in the correct format) you want to keep, it should only take a few minutes to edit the note file by hand, unless you are an awfully busy person.

When prompted for input, the format for the dates in MM/DD/YY and for the hours it is HHMM. Within the note file, the "short dates" of month and day — 1-9 — are padded in front with zeros to keep consistency with the format the Model 100 uses for its Date$. But it isn't necessary to supply the extra zeros when you are prompted for input dates; as with the colons, the program will add them for you. (You can slightly simplify the program by removing the lines which use the note file in that format.)

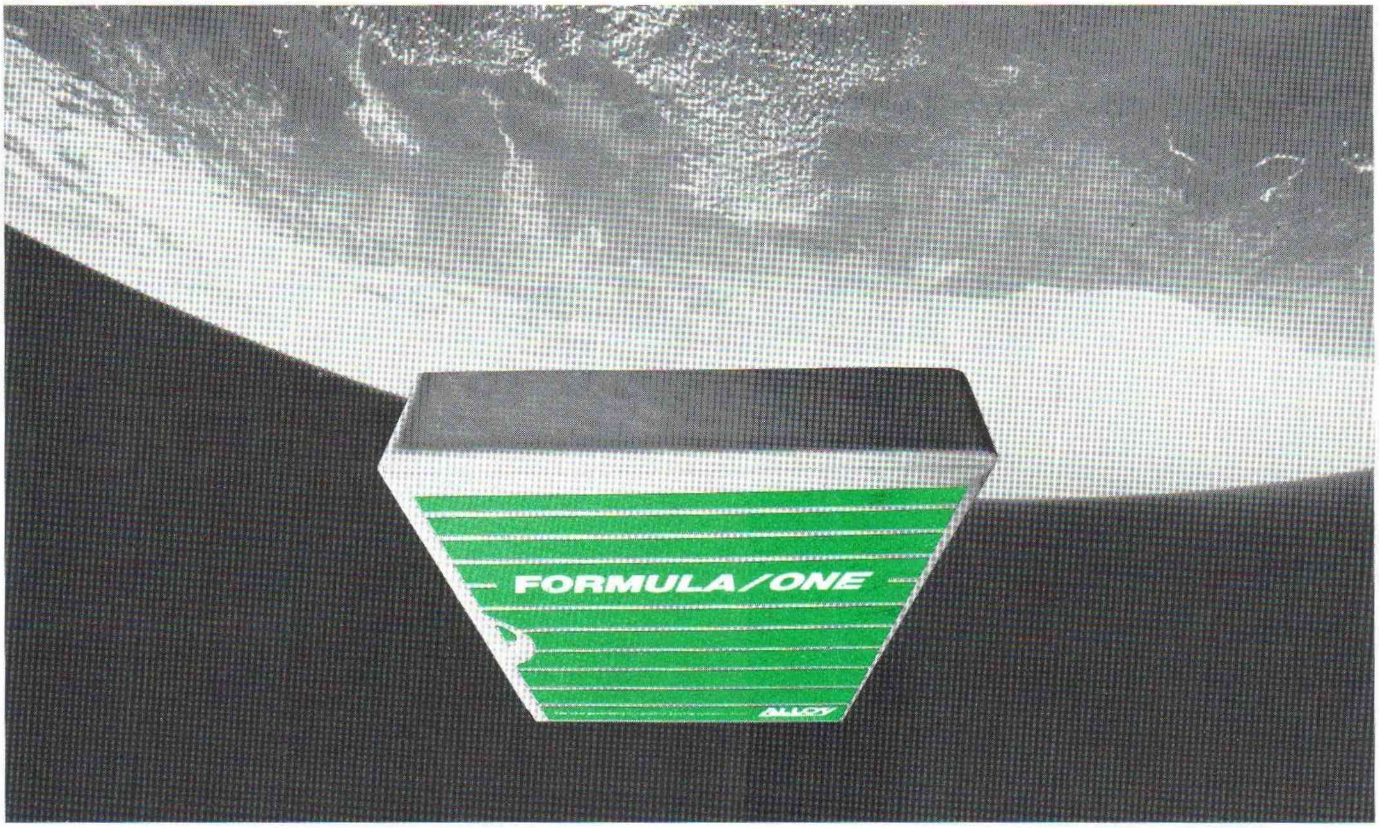
I require the size of the details of an appointment kept down to a short string, less than 27 characters, so that one entire appointment record occupies one Model 100 screen line length of 40 characters. (Incidentally, you could change the format to YY/MM/DD for the dates. This, with the necessary alterations to the sort keys used in the program, would slightly reduce the program in size — a few lines — and very slightly improve its speed. Speed in performance hasn't been a problem, though, with moderately-sized note files.) The program does some error-checking on the input dates and times to see that they are in the right format.

Sample Screen



```
Sat   Sun   Mon   Tue   Wed   Thu   Fri
1     2     3     4     5     6     7
8     9     1Ø    11    12    13    14
15    16    17    18    19    2Ø    21
22    23    24    25    26    27    28
29    3Ø    31
Jun Ø1,1985 Mon 18:54    JUNE,1985
Clnd Dail Appt  Del Alrm          Menu
```

(Jay Halcomb is just completing his Ph.D. in Philosophy at the University of Arizona. Specializing in mathematical logic and the philosophy of mathematics, he has been a Model 100 user and programmer for over two years.)



The fastest equation solver on earth can win you an IBM PC AT.

Introducing FORMULA/ONE, the world's fastest and most versatile equation solving tool for use on your personal computer. No matter how you are solving equations now, FORMULA/ONE can beat it. FORMULA/ONE will make short work of your toughest engineering problems. And best of all, it doesn't require any programming. So you're free to use your time analyzing results, not producing them.

Free demo diskette. No other product can match the range and functionality of FORMULA/ONE. And no other product has all these features: Multiple regression analysis, curve fitting, 47 intrinsic functions, integration with your data base, compatibility with Lotus 1-2-3, Supercalc, and Visicalc, solves linear, non-linear, simultaneous and transcendental equations and graph plotting. To show you just what FORMULA/ONE can do for you, we'll send you a free demo diskette that explains it all.

Win an IBM PC AT. When you send for the free demo diskette, you'll also become eligible to win an IBM PC AT.

If you want FORMULA/ONE—the fastest equation solver on earth right now, call (617) 875-6100.

Enter me for an IBM PC AT and send me the free FORMULA/ONE demo diskette. All entries will be eligible to win. The winner will be selected at random.

NAME: ________________________________

POSITION: ________________________________

COMPANY: ________________________________

ADDRESS: ________________________________

____

PHONE: ________________________________

ALLOY

Mail to:
FORMULA/ONE CONTEST
Alloy Computer Products
100 Pennsylvania Ave.
Framingham, MA 01701

To enter use an official entry form or reasonable facsimile. Form may not be duplicated after completion. You must be 18 years or over to enter. All information must be complete to be eligible.

No purchase necessary. There will be one winner determined from the first entry drawn at random. Entries for the contest must be postmarked no later than August 15. The drawing will be held on August 20. The winner will be contacted by phone and mail.

Any taxes associated with claiming the prize are the responsibility of the winner. Cash value may not be substituted for the prize. Employees of Alloy Computer Products, Inc. and their agencies are not eligible. The decision of the contest judges is final.

Prize is non-transferable. This promotion is void where prohibited or restricted by law.

The time (HHMM) is based on a 24-hour military clock, like the Model 100 uses. It would require a bit more complexity in the program to convert these to the more usual 12-hour a.m. and p.m. times. I didn't feel it was worth the extra space the program would require to do this, but it is one way in which the program could be customized.

Since the program is completely modularized, it should be a simple task to customize it to your own tastes. You might want just the calendar feature, for instance. The program, with remarks, occupies a little over 7K of RAM; removing the remarks will bring that down to a bit over 4K. The program could be further reduced by another 1K or so by removing some of the error-checking that is done on the inputs (if you have confidence in your typing).

Besides these space requirements, you will also have to have available free space to equal the size of your note file when it is being duplicated, to make and delete appointments in it.

The trickiest part of the program is the Calendar function. The difficulty in displaying a calendar is that you have to know what day of the week the first day of the month falls on. I calculate this in the program by a simple perpetual calendar formula, then use the `Print at` command to get the calendar dates printed under the right days of the week.

The calendars are valid for any months in the 20th century. If you want calendars for other dates, the formula I use can be easily modified. (The source of the formula was *Webster's Dictionary*, third edition, perpetual calendars.)

A cautionary note: If you panic out of the program with a CONTROL-C or SHIFT-BREAK, you will leave the function keys defined as in the program. Then, you will want to either make the machine calls necessary to reset them manually from BASIC, or (easier) re-enter the program and exit normally. Also, you want to be careful in using the function keys; if you inadvertently hit two, the program will progress through two modes rapidly. This won't harm anything, but might be an annoyance.

A guide to the program listing follows. In the interest of conserving space in the listing and in RAM, I've followed the common practice of combining a number of lines into one, using the ":" contraction. (Caution: You can't do this with lines containing the `IF/THEN` construction, unless it is the last item in the line.)

PCM BAR CODED LISTING

The listing:

```
10 'Schedl.ba -- Model 100 appointments
program for use with "Note.do".
20 'Functions: display calendar with app
't's, enter app't's in note file,
30 '     delete app't's, display daily ap
p't's, set alarm, exit to menu.
40 'Usage is through function keys and p
rompted input. Input is:
50 '     Date for calendar display:
MM/YY            (digits)
60 '     Date for daily app't's:
MM/DD/YY             "
70 '     Date to enter/del. app't:
MM/DD/YY             "
80 '     Hour of app't:
HHMM                 "
90 '     Details of app't:
string           (len < 27)
100 '    Time for alarm:
HHMM             (digits)
110 '    Message for alarm:
string
120 'It is unnecessary to pad input date
s with leading 0's; they will be
130 '    inserted.
140 'Some, but not complete, checking is
 done for correct format of input.
150 'The program assumes the note file i
s sorted, in the following format:
160 '    MM/DD/YY:HHMM:<details of app't>
. The sort order, least first, is:
170 '    first by year, then by month, th
en by day, then by hour and minutes.
180 'Dates in the note file are assumed
to be in, and are maintained
190 '    in, the standard Model 100 forma
t: padded with leading 0's where
200 '    necessary. Time is in 24 hr. clo
ck (military). The sort order is
210 '    maintained.
```

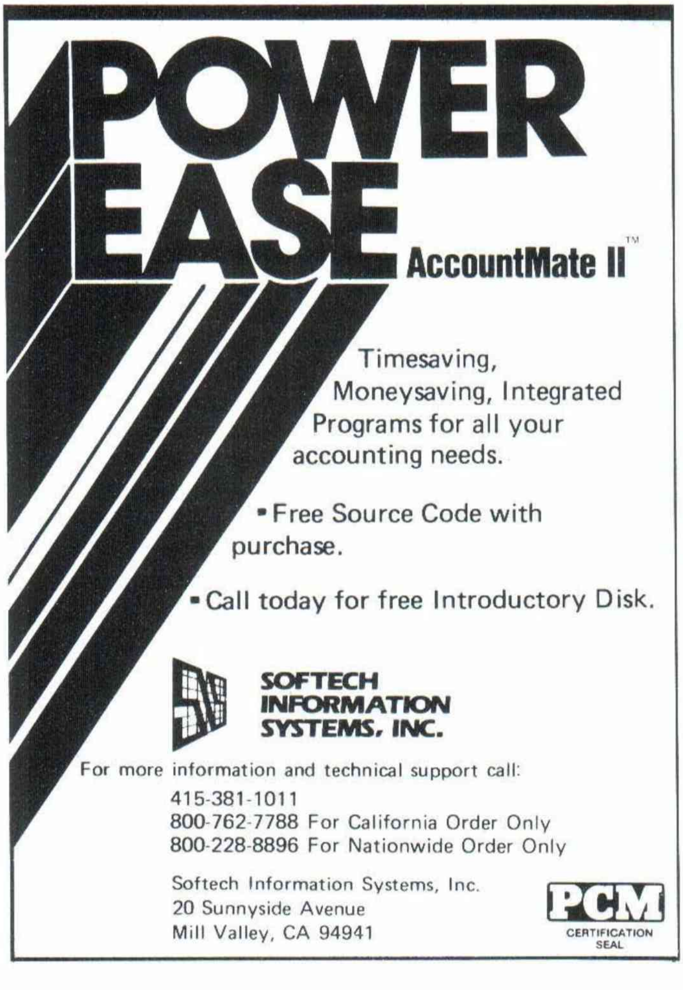
Tour of the listing:

Lines	**Function**
10-300	Explanatory remarks
310-460	This is the main driver. We dimension the necessary arrays and read the data. This consists of the names of the months, followed by the number of days in each, followed by a jigger factor used in the perpetual calendar formula. The names go into the array M$, the number of days into M, the jigger factors into MN; the array D is used later to store the dates of appointments in a particular month (all of these used in Calendar mode). The array D$ (used in the Daily mode) will hold the appointment details for a particular date. In Line 380, we `GOSUB` the calendar module upon first entering the program to display the calendar for the current month. In 400-430, we define the function key labels, turn them on and tell the program where to go when the right function key is pressed. In 450-460, we loop until a function key is pressed, beeping if some unwanted key is pressed. Depending on the key pressed, we `GOSUB` in Line 400 to the appropriate module of the program.
490-670	This begins the Calendar module. We input the month to display, in the format MM/YY. In 520, if ENTER is pressed, we read the

```
220 'The program takes account of leap y
ears in computing the calendar.
230 'The calendar is invalid before 1900
 and beyond 2000 a.d.
240 'Pressing "Enter" in the Calendar or
 Daily modes assumes current date.
250 'Pressing "Enter" in the App't, Dele
te, or Alarm modes returns to
260 '    main function selection.
270 '
280 '
290 'Main driver, initialization, and da
ta.
300 '    Initialization and data.
310 CLS:MAXFILES=2:SCREEN 0,0:DIM M$(12)
:DIM M(12):DIM D(31):DIM D$(25):DIM MN(1
2):ON ERROR GOTO 310
320 FOR I=1 TO 12:READ M$(I),M(I),MN(I):
D$(I)="":NEXTI:D$="":H$="":DT$=""
330 DATA JANUARY,31,1,FEBRUARY,28,4,MARC
H,31,4,APRIL,30,0,MAY,30,2,JUNE,31,5,JUL
Y,31,0,AUGUST,31,3,SEPTEMBER,30,6,OCTOBE
R,31,1,NOVEMBER,30,4,DECEMBER,31,6
340 IF VAL(MID$(DATE$,7,2)) MOD 4<>0 THE
N 370
350 '    Reset for leap years
360 M(2)=29:MN(1)=0:MN(2)=3
370 '    Display current calendar first t
```

current month from the Date$. In 540-550, we check to see that the D(ate)$ is the right length and has the right numbers (no month 46, e.g.!). If something's wrong, we inform the user in 560. 570 tells us the value of the month and year sort key we are going to match in the Note.do file is the value of the month plus 100 times the value of the year. Next, we clear the input messages from the screen and print the banner with the days of the week accross the screen, using Print at 0 to begin printing at the top left corner. In 600-670, we read through the note file, comparing the values of the month and year we find there in each appointment record with the one we calculated in 570. If the read values are too small, we keep reading; if we find a match, we make a record of it in the D array. When the values get too big, we stop looking and close the file.

690-800 This completes the Calendar module. Here we print the Calendar and return to the main selection function, leaving the Calendar on the screen. Line 700 contains the formula for the perpetual calendar: (YY + INT(YY/4) + jigger factor +1)/7. The remainder of this expression is the day of the week that the first of the month fell on, for that month and year, and beginning with Saturday as the zeroth day. This number is multiplied by six to get the number of spaces over on the



```
ime through.
380 GOSUB 520
390 '    Define function keys and labels.
400 KEY 1,"Clnd":KEY 2,"Dail":KEY 3,"App
t":KEY 4," Del":KEY 5,"Alrm":KEY 6,"":KE
Y 7,"":KEY 8,"Menu"
410 KEY(1)ON:KEY(2) ON:KEY(3) ON:KEY(4)
ON:KEY(5) ON:KEY(8) ON
420 ON KEY GOSUB  500,840,1110,1470,1750
,,,1890
430 SCREEN 0,1
440 '    Loop here for function key input
.
450 B$=INKEY$:IF B$="" THEN 450:
460 BEEP:GOTO 450:
470 '
480 'Module -- display calendar
490 '    Get month to display, print bann
er.
500 CLS :PRINT "Calendar:":INPUT "Month/
year";D$
510 '              If "Enter" pressed, assu
me current month.
520 IF D$="" THEN D$=MID$(DATE$,1,2)+"/"
+MID$(DATE$,7,2)
530 '              Check for correct format
: MM/YY.
540 IF INSTR(1,D$,"/")=0 THEN 560
```

screen to begin printing the dates. Line 710 contains a switch — we want to print a little banner with the current date and time, but there isn't going to be room on the screen for it on a line by itself, so we print it at the top or bottom of the calendar, wherever there is space, depending on which day of the week the first fell on that month and how many days there are in that month. In Line 720, we print the banner by PEEKing at the place in the Model 100 RAM where that information is stored in ASCII. In lines 730-780, we print the dates, beginning at the previously computed displacement (the variable D). Here, it's just a matter of counting over six spaces, printing the next date and keeping track of when to begin a new line (Line 760 does this). In 790, we print the name of the month and the year at the bottom right of the calendar. Notice that when we were printing the dates in 790, we checked our scratch-pad D array to see if we had made note of an appointment on that date — if we had, we print the date in brackets. Finally, in 800, we "re-zero" our array and other counters, and return to 450 where we were when the On Key Gosub interrupted us.

820-1070 This is the Daily module. Here we get the date to search for in the note file (in Line 890 we GOSUB 1920 — this is a subroutine that checks that the D$ is in the right format,

```
550 IF VAL(LEFT$(D$,INSTR(1,D$,"/")-1))<
13 AND (MID$(D$,2,1)="/" OR MID$(D$,3,1)
="/") AND (LEN(D$)=4 OR LEN(D$)=5) THEN
570
560 PRINT "Error in input:":BEEP:PRINT D
$:FOR I=1 TO 1000:NEXT I:CLS:D$="": GOTO
 400
570 V=VAL(LEFT$(D$,INSTR(1,D$,"/")-1))+1
00*VAL(MID$(D$,INSTR(1,D$,"/")+1))
580 CLS:PRINT @0," Sat   Sun   Mon   Tue
   Wed   Thu   Fri";
590 '    Read note file for app'ts.
600 OPEN "note.do" FOR INPUT AS #1
610 IF NOT EOF(1) THEN LINE INPUT #1,I$
620 W=VAL(MID$(I$,1,2))+100*VAL(MID$(I$,
7,2))
630 IF W<V AND NOT EOF(1) THEN 610
640 IF W>V THEN 670
650 IF W=V THEN D(VAL(MID$(I$,4,2)))=1
660 IF NOT EOF(1) THEN 610
670 CLOSE
680 '    Compute and display calendar, re
turn.
690 M=VAL(LEFT$(D$,INSTR(1,D$,"/")-1))
700 D=6*((VAL(MID$(D$,INSTR(1,D$,"/")+1)
)+INT(VAL(MID$(D$,INSTR(1,D$,"/")+1))/4)
+MN(M)+1) MOD 7)
710 IF M(M)>=30 AND D>=30 THEN X=40 ELSE
 X=240
720 FOR I=0 TO 20:PRINT @X+I,CHR$(PEEK(6
4904!+I));:NEXT I
730 P=40:N=1
740 IF D(N)=1 THEN PRINT @P+D,"["+RIGHT$
(STR$(N),LEN(STR$(N))-1)+"]"; ELSE PRINT
 @P+D,N;
750 N=N+1:P=P+6
760 IF (P+D) MOD 40<>2 THEN 780
770 P=P+D-2:D=0
780 IF N<M(M)+1 THEN 740
790 PRINT @265,M$(M)+",19"+MID$(D$,INSTR
(1,D$,"/")+1);
800 FOR I=1 TO 31:D(I)=0:NEXT I:M=0:D$="
":RETURN
810 '
820 'Module -- display daily app't's.
830 '    Get date to display, check for c
orrect format: MM/DD/YY.
840 CLS:PRINT "Daily:":INPUT "Date";D$
850 '              If "Enter" pressed, assu
me current date.
860 IF D$="" THEN D$=DATE$
870 IF MID$(D$,2,1)="/" THEN D$="0"+D$
880 IF MID$(D$,5,1)="/" THEN D$=MID$(D$,
1,3)+"0"+MID$(D$,4)
890 GOSUB 1920:CLS
900 '    Read note file for app't's.
910 OPEN "note.do" FOR INPUT AS #1
920 V=10000*VAL(RIGHT$(D$,2))+100*VAL(LE
FT$(D$,2))+VAL(MID$(D$,4,2)):N=1
```

see below), search for appointments on that day as we did in the Calendar module, and storing those found in the array D$ which we print from in 1000. If we print more than five, the screens begins to fill up, so in 1030, if we have more than a screen full to print, we ask for instructions to start printing a new screen after erasing the previous one. Finally, we return from the interrupt, leaving the appointments found still on the screen.

1110-1430 This is the Appointments module. We get the information needed to make the appointment (date: MM/DD/YY, time: HHMM, details). In the Line 1160, we GOSUB 1920 to test the format of the D$; in Line 1180, we GOSUB 2010 to a subroutine that tests the format of the H$. Next, we open and begin reading the note file until we come to a date past the one on which the appointment is to be made. This is where we want to insert the new appointment (Line 1330). But, instead of just reading the note file, we've been copying it at the same time to the "temp" file opened in 1240, so after inserting the new appointment, we copy the rest of the old note file on top of it in the temporary file, kill the old note file and rename the temp file to note.do (Line 1380). Then we return after confirming the appointment (1400-1410). (If we found we already had an appointment at that time, we go to 1430 to display this fact, and return.)

1470-1710 This is the Deletion module. After getting the time and date of the appointment to delete and verifying them, we copy the note file to a temporary file, as we did in the Appointment module, but we don't copy the appointment to be deleted. Then, we erase the note file and name the temp file to be the new note file. A switch (Line 1670) kept track that we actually did find an appointment with the right time and date; if we didn't, we go to 1680 and advise the user. Otherwise, we advise the user in 1710 and return.

1750-1850 This is the Alarm module. We get the time to set the alarm for (format: HHMM, which we verify by GOSUBing 2010). Then, we use the Model 100's On Time$= interrupt instruction in Line 1810. When Time$=T$, the 100 will stop whatever it is doing (in the program) and GOSUB 1850, where we beep at the user and display whatever message he wanted to see at that time. Incidentally, we confirm the time set and return.

1890 The Menu module. We make the appropriate machine calls to turn off the function keys we've been using, to restore them to the normal BASIC mode, then exit to the Menu.

1920-1980 This subroutine verifies that the D(ate)$ is in the correct format (MM/DD/YY), and returns to whatever module called it with a GOSUB. If there is an error in the D$, we advise the user of it in 1980 and restart the program.

2010-2070 Similarly, this subroutine verifies the H(our)$: HHMM.

```
930 IF NOT EOF(1) THEN LINE INPUT #1,I$
940 W=10000*VAL(MID$(I$,7,2))+100*VAL(MI
D$(I$,1,2))+VAL(MID$(I$,4,2))
950 IF W>V THEN 990
960 IF W<V AND NOT EOF(1) THEN 930
970 IF W=V THEN D$(N)=MID$(I$,10):N=N+1
980 IF NOT EOF(1) THEN 930
990 CLOSE:N=N-1:C=0
1000 '  Display app't's found, return.
1010 PRINT @0,D$+":":FOR I=1 TO 5:PRINT
D$(I+C):NEXT I:C=C+5:IF C>= N THEN 1070
1020 '           Prompt for additional di
splay if screen full.
1030 PRINT @240,"More (Y/N)?";
1040 A$=INKEY$:IF A$="" THEN 1040
1050 PRINT @240,"           ";:IF A$="N"
 OR A$="n" THEN 1070
1060 CLS:GOTO 1010
1070 :FOR I=1 TO 25:D$(I)="":NEXT I:D$="
":RETURN
1080 '
1090 'Module -- enter app't.
1100 '  Input date, hour, details. Check
 formats: MM/DD/YY, HHMM.
1110 CLS:PRINT "Make appointment:":INPUT
 "Date";D$
1120 '           If "Enter" pressed, retu
rn.
1130 IF D$="" THEN 1410
1140 IF MID$(D$,2,1)="/" THEN D$="0"+D$
1150 IF MID$(D$,5,1)="/" THEN D$=MID$(D$
,1,3)+"0"+MID$(D$,4)
1160 GOSUB 1920
1170 INPUT "Hour";H$
1180 IF H$="" THEN 1410
1190 GOSUB 2010
1200 INPUT "Details";DT$
1210 IF DT$="" THEN 1410
1220 IF LEN(DT$)<27 THEN 1250
1230 BEEP:PRINT "Error in input:":PRINT
"Details too long":FOR I=1 TO 1000:NEXT
I:D$="":H$="":DT$="":CLS:GOTO 400
1240 '  Copy note file to temp file, ins
ert app't, rename temp file.
1250 CLS:OPEN "note.do" FOR INPUT AS #1:
OPEN "temp.do" FOR OUTPUT AS #2
1260 V=VAL(H$)+10000*VAL(MID$(D$,4,2))+1
E+08*VAL(MID$(D$,1,2))+1E+11*VAL(MID$(D$
,7,2))
1270 IF NOT EOF(1) THEN LINE INPUT #1,I$
1280 W=VAL(MID$(I$,10,4))+10000*VAL(MID$
(I$,4,2))+1E+08*VAL(MID$(I$,1,2))+1E+11*
VAL(MID$(I$,7,2))
1290 IF W<V THEN PRINT #2,I$
1300 IF W<V AND NOT EOF(1) THEN 1270
1310 '           Check for duplicate app'
```

```
t.
1320 IF V=W THEN 1430
1330 PRINT #2,D$+":"+H$+":"+DT$:IF W>V T
HEN PRINT #2,I$
1340 IF EOF(1) THEN 1380
1350 IF NOT EOF(1) THEN LINE INPUT #1,I$
1360 PRINT #2,I$
1370 IF NOT EOF(1) THEN 1350
1380 CLOSE 1:CLOSE 2:KILL "note.do":NAME
 "temp.do" AS "note.do"
1390 '  Confirm app't. Return.
1400 BEEP:PRINT "Appointment made:":PRIN
T D$+":"+H$+":"+DT$:FOR I=1 TO 1000:NEXT
 I
1410 D$="":H$="":DT$="":CLS:RETURN
1420 '  Display duplicate app't. Return.
1430 CLS:BEEP:PRINT "You already have an
 appointment at this time:":PRINT I$:FOR
 I=1 TO 1000:NEXT I:CLOSE 1:CLOSE 2:KILL
 "temp.do":CLS:D$="":H$="":DT$="":RETURN
1440 '
1450 'Module -- delete app't from note f
ile.
1460 '  Input date, hour. Check formats:
 MM/DD/YY, HHMM.
1470 CLS:PRINT "Delete Appointment:":INP
UT "Date";D$
1480 '            If "Enter" pressed, retu
rn.
1490 IF D$="" THEN 1690
1500 IF MID$(D$,2,1)="/" THEN D$="0"+D$
1510 IF MID$(D$,5,1)="/" THEN D$=MID$(D$
,1,3)+"0"+MID$(D$,4)
1520 GOSUB 1920
1530 INPUT "Hour";H$
1540 IF H$="" THEN 1690
1550 GOSUB 2010
1560 '  Copy note file to temp file, del
ete app't, rename temp file.
1570 CLS:SW=0
1580 V=VAL(H$)+10000*VAL(MID$(D$,4,2))+1
E+08*VAL(MID$(D$,1,2))+1E+11*VAL(MID$(D$
,7,2))
1590 OPEN "note.do" FOR INPUT AS #1:OPEN
 "temp.do" FOR OUTPUT AS #2
1600 IF NOT EOF(1) THEN LINE INPUT #1,I$
1610 W=VAL(MID$(I$,10,4))+10000*VAL(MID$
(I$,4,2))+1E+08*VAL(MID$(I$,1,2))+1E+11*
VAL(MID$(I$,7,2))
1620 IF V=W THEN 1710
1630 PRINT #2,I$
1640 IF NOT EOF(1) THEN 1600
1650 CLOSE 1:CLOSE 2:KILL "note.do":NAME
 "temp.do" AS "note.do"
1660 '  Check that deletion was made. If
 not, advise, return.
1670 IF SW=1 THEN 1690
1680 BEEP:PRINT "Not found:":PRINT D$+":
"+H$:FOR I=1 TO 1000:NEXT I
1690 CLS:D$="":H$="":RETURN
1700 '  Confirm deletion. Return.
1710 PRINT "Deleting:":BEEP:SW=1:PRINT I
$:FOR I=1 TO 1000:NEXTI:CLS: IF NOT EOF(
1) THEN 1600 ELSE 1650
1720 '
1730 'Module -- set alarm.
1740 '  Input time, check format: HH:MM:
SS. Set timer.
1750 CLS:INPUT "Time to set";H$
1760 '            If "Enter" pressed, retu
rn.
1770 IF H$<>"" THEN 1790
1780 CLS:RETURN
1790 GOSUB 2000
1800 INPUT "Message";M$:T$=MID$(H$,1,2)+
":"+MID$(H$,3,2)+":00"
1810 ON TIME$=T$ GOSUB 1850:TIME$ ON
1820 '  Confirm, return.
1830 BEEP:PRINT "Time set:":PRINT T$:FOR
 I=1 TO 1000:NEXT I:CLS:H$="":RETURN
1840 '  Here is the alarm. Turn timer of
f. Return.
1850 FOR I=1 TO 10:BEEP:PRINT M$:PRINT:N
EXT I:CLS:T$="":TIME$ OFF:RETURN
1860 '
1870 'Module -- exit to Menu.
1880 '  Turn off function keys.
1890 CALL 23164,0,23366:CALL 27795:MENU
1900 '
1910 'Subroutine -- test D(ate)$ for cor
rect format: MM/DD/YY. Return.
1920 IF LEN(D$)<>8 OR MID$(D$,3,1)<>"/"
OR MID$(D$,6,1)<>"/" OR VAL(MID$(D$,1,2)
)>12 OR VAL(MID$(D$,4,2))>31 THEN 1980
1930 CK$=MID$(D$,1,2)+MID$(D$,4,2)+MID$(
D$,7,2)
1940 FOR I=1 TO 6
1950 IF ASC(MID$(CK$,I,1))<48 OR ASC(MID
$(CK$,I,1))>57 THEN 1980
1960 NEXT I
1970 RETURN
1980 BEEP:CLS:PRINT "Error in input:":PR
INT D$:FOR I=1 TO 1000:NEXT I:D$="":H$="
":DT$="":CLS:GOTO 400
1990 '
2000 'Subroutine -- check H(our)$ for co
rrect format: HHMM. Return.
2010 IF LEN(H$)<>4 THEN 2070
2020 IF VAL(MID$(H$,1,2))>23 OR VAL(MID$
(H$,3,2))>59 THEN 2070
2030 FOR I=1 TO 4
2040 IF ASC(MID$(H$,I,1))<48 OR ASC(MID$
(H$.I,1))>57  THEN 2070
2050 NEXT I
2060 RETURN
2070 BEEP:PRINT "Error in input:":PRINT
H$:FOR I=1 TO 1000:NEXT I:D$="":CLS:H$="
":GOTO 400
```

Changes your Model 100 into a totally different computer with capability you never thought possible.

Lucid™ on Snap-In™ ROM

Infoworld rated Lucid's performance "excellent"

PCSG says "Satisfaction Guaranteed or your money back within 30 days!"

LUCID© is here now. It is on a ROM cartridge that snaps into the compartment on the back of your Model 100. It takes no memory to load and no memory for operating overhead. That means you have the full 29.6k bytes free to store your data.

First, LUCID© is memory conserving. It will let you build a large spreadsheet—255 row by 126 column capacity. You build huge spreadsheets in your Model 100's RAM that could consume 80 to 100K on a desktop computer.

Secondly, LUCID© is fast. LUCID© is so rapid, a 36 column corporate financial statement took less than 4 seconds to calculate.

Thirdly, LUCID© has features you won't find in most other spreadsheets. For example, when you type a label (text) it will cross column boundaries; in other words when you type a label or title it will appear as you type it irrespective of column or width. LUCID© also allows you to set column widths individually, and of course LUCID© has insert row and insert columns, as well as other standard features. LUCID© even lets your formulas refer to cells in other spreadsheet files.

Further, LUCID© has what no other spreadsheet has: Cut, Copy, and Paste. It uses the same keys as Cut and Paste in TEXT, but here's the difference: it takes all the formulas with it when you paste and they all automatically recalculate with the entire sheet.

And here is what is really amazing. You can copy or cut from one spreadsheet and paste into another spreadsheet or even a TEXT file.

LUCID©supports all BASIC math functions as well as Log, sine, cosine, tangent, exponentiation and other sophisticated math functions.

LUCID© has so many features that you will say "this is what I need in a spreadsheet", such as automatic prompting of an incorrectly typed-in formula showing just where the mistake was made.

LUCID© has expanded "go to" functions that remember and produce a windowing capability.

But perhaps most remarkable is that LUCID© is not only a spreadsheet but a program generator as well. First, LUCID© lets you protect all cells against entry or change, and then unprotect just the cells you want for someone else to use as input fields.

LUCID© will not only process values, but text input as well so that the facts other than numbers can be responded to. LUCID© has the ability for you to refer in a formula to cells containing words. This feature combines with the capacity of doing "if then" statements that work by doing table lookups against even massive X/Y charts of text or numerical information. You can produce a program that responds to inputs with no programming knowledge whatsoever.

You can prepare a report section in your spreadsheet with instructions to your user for printout, and they can produce a personalized printout that responds to their input. All your formulas and tables that did the calculations and provided the facts are invisible to that user. LUCID© is useful for doctors for patient questionnaires, troubleshooting technicians, purchase clerks, people doing job quotes, stores for customer workups, insurance agents and anybody who needs to process specific facts and numbers to produce a report based on those responses.

LUCID© comes with a manual that explains not only the characteristics of LUCID©, but will train you how to use a spreadsheet even if you have never seen one before. You are shown how to do budgets, forecasts, breakeven analysis amortizations and many other types of personal and business reports and calculations.

User friendly is such an over-used term in this industry, but a typical comment has been "I have never seen a spreadsheet that does so much, and yet LUCID© is so much easier and faster to use."

LUCID© is a result of a most exhaustive developmental effort in which PCSG's objective was to develop a spreadsheet that was better than the state-of-the-art. We are so pleased because LUCID© provides for the Model 100 spreadsheet capability you cannot equal on a desktop computer.

LUCID© is, in our opinion and that of those who have examined it, a breakthrough. We sell it on a 30 day trial. If you are not completely satisfied, return it within 30 days for a full refund. Priced at $149.95, on snap-in ROM. MasterCard, Visa or COD.

© PCSG 1984

First with software for the Model 100

PORTABLE
COMPUTER
SUPPORT
GROUP

1-214-351-0564

11035 Harry Hines Blvd., #207
Dallas, Texas 75229

Programming with dBASE II

Once You Get the Picture What do You Say?

By Danny Humphress
PCM Technical Editor

In the June edition of dBASE Tutor, we used FMT files to design nice-looking data input screens. We also learned how to use the PICTURE clause of the GET statement to format fields as they are entered. Today, we'll do a little more with PICTURE and learn about a similar clause for the SAY statement.

Popular Pictures

Using the five picture function symbols, #, 9, A, ! and X, we can create a great number of specialized GET templates. Here are some of the more common PICTURE clauses:

`'999-99-9999'`	Social Security Number
`'(999)999-9999'`	Telephone Number
`'!!'`	State Abbreviation
`'9999.99'`	Dollar Amount
`'99999-9999'`	ZIP + Four
`'99/99/99'`	Date

Keep in mind that the user of your data entry screen will not be able to type over or change any of the nonfunctional symbols. When you use the telephone number template, for example, the user will only have to type the digits. The parentheses and the dash will stay put. The only exception is the decimal point (.) when used for entering numbers. The figure you enter will automatically align with the decimal point when you press ENTER.

Say Using

Just as GET's PICTURE clause allows you to control the way data is entered into the file, SAY has a similar clause, USING, which controls the way fields and variables appear on the screen.

The format for SAY USING is:

```
@ x,y SAY z USING template
```

Where x,y are screen coordinates, z is a numeric expression such as a variable, field, constant, or a formula, and template is a string made up of special symbols.

The special symbols used with SAY USING are very similar to those used with GET PICTURE:

— Prints a digit only
9 — Same as 9
X — Prints any character
$ — Prints a digit or a $
* — Prints a digit or a *
, — Prints a comma in numbers
. — Aligns decimal point

The # and 9 symbols work with both string and numeric fields/variables, X works only with strings, and the other symbols apply to numbers.

Here's how the number, 12345.67, would appear with different picture clauses:

1.	`'    9999999'`	`12345`
2.	`'   99999.99'`	`12345.67`
3.	`'  99999.999'`	`12345.670`
4.	`'  99,999.99'`	`12,345.67`
5.	`'$999,999.99'`	`$ 12,345.67`
6.	`'$$$$,$$$.99'`	`$$12,345.67`
7.	`'$***,***.99'`	`$*12,345.67`
8.	`'     999.99'`	`345.67`

(Danny Humphress, PCM's technical editor, is the owner of a computer software and consulting firm in Louisville, Ky. Danny brings to PCM his extensive experience with small business computers and applications software.)

USE YOUR DESKTOP COMPUTER AS A DISK DRIVE FOR YOUR M-100.

Disk+

on Snap-In™ ROM

Try *Disk*+ for 30 days. If you aren't as excited as we are, return it for a full refund.

When we designed *Disk*+ we did it out of necessity. We wanted a way that we could just connect a Model 100 to our desktop computer with a cable and save files onto the desktop's disk drive. We wanted it to be so simple to use it would be self-explanatory.

Picture this. *Disk*+ comes to you on a Snap-in ROM and a diskette for your desktop. You take a quarter and open the little compartment on the back of your Model 100. Then you just press the ROM into the socket. *Disk*+ appears on your main menu just like a built-in.

You connect your Model 100 to your other computer using an RS232 cable (available from PCSG for $40).

You just place the *Disk*+ diskette into the desktop's drive and turn on the computer. It powers up automatically and says "awaiting command" on your desktop's screen. Then you just put the widebar cursor on the Model 100 main menu on *Disk*+ and press ENTER. You are shown your RAM files arranged just like the main menu.

To save a file to your other system's disk drive, you just move the widebar cursor to the file you want to save and press ENTER. It is saved instantly with no further action.

To look at the disk directory, you just press a function key on your Model 100. You see immediately the disk directory on your Model 100 screen, and it is arranged just like your Model 100's main menu.

To load a file from the diskette to your Model 100, you just move the widebar cursor to the file and press ENTER. The file is transferred to your Model 100's RAM instantly. You can press F8 and go back to the main menu, and the file you loaded from diskette is there, ready to use.

It is so nice to be able to keep your documents, programs (both BASIC and machine code) and *Lucid* spreadsheet files on the diskette, and bring them back when you need them. All files are ready to run or use with no changes or protocol by you.

If you have access to a desktop computer and don't have *Disk*+, then evidently we have done a poor job telling you about it.

All files and programs that you load or save, go over and come back exactly as they are supposed to be because of full error checking. This guaranteed integrity is really a comfort. *Disk*+ is wonderful in so many other ways. For example, you can do a "save all" of all your RAM files with just a touch of a function key. That group of files is saved on the diskette under a single filename with a .SD (for subdirectory) extension. Any time you want, you can bring back all those files at once, or just one or two if you like, again with one-button ease.

Disk+ takes up no RAM. That's zero bytes either for storing the program or for operating overhead.

What really excites most *Disk*+ users is text file cross compatibility. Your Model 100's text files are usable on your desktop computer, and your desktop's text files become Model 100 text files.

This means you can write something on your Model 100, and with *Disk*+ transfer it instantly to your desktop and start using it right away on your bigger computer. Or the way we like to work is to type in a document on the desktop computer and then transfer it to our Model 100 with *Disk*+. Then we print out the document, beautifully formatted, using WRITE ROM.

1-214-351-0564

Disk+ works with just about every micro sold, from IBM PC and its clones, to all Radio Shack computers (yes, all), to Apple II, Kaypro, Epson and most CPM. Just ask us. More than likely, your computer is supported.

Incidentally, hundreds of Model 100 owners have gone to their Radio Shack stores and bought a color computer because it is so low priced, and with *Disk*+ they have an inexpensive disk drive.

And if that weren't enough, how about this: *Disk*+ also provides cross-compatibility between different computers like IBM, Apple or the Model 4 using the Model 100 as the intermediary device. Quite a feature!

The snap-in ROM is really great because you can use other ROMs like *Lucid* or WRITE ROM. They snap in and out as easily as an Atari game cartridge and you never lose your files in RAM.

Anyone who ever uses *Disk*+ simply can't do without it. But so many times we have had new users call us and say, "Wow! I had no idea when I ordered it that *Disk*+ would be so fantastic. I just couldn't believe that I could use my desktop computer's disk drive with my Model 100 just like it is another main menu."

That's why we sell *Disk*+ on a thirty-day trial. If you aren't completely satisfied, return it within thirty days for a full refund. Priced at $149.95 on Snap-in ROM. MasterCard, Visa or COD.

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd., #207, Dallas, Texas 75229 • 1(214) 351-0564

PCSG provides hotline software support for the Model 100. Call us at 1-214-351-0564

In the first template, we did not provide a decimal place, therefore, the fractional part of the number was truncated (dropped).

When we added the decimal point in the second and third examples, the appropriate number of decimal places were printed.

In the fourth example, we added a comma to separate the thousands place. If the number had been smaller than 1,000, the comma would not have printed.

A dollar sign was added to the fifth example to make the number look like a monetary figure. In this example, however, there is a space between the dollar sign and the first digit. dBASE doesn't have a "floating" dollar sign function built in, but we can have the dollar sign repeat as is demonstrated in the next example.

The seventh example shows a variation of the two previous ones. It fills the space between the dollar sign and first digit with asterisks. This is handy for use in programs that print checks (more on that later). Notice that we also used the comma in these examples. If the number had been smaller than 1,000, an asterisk (or dollar sign) would have been printed in place of the comma.

The final example demonstrates a problem which you must be careful of when programming with dBASE II. Since there weren't enough digit places to the left of the decimal point, the '1' and '2' were just cut off. You just lost $12,000! It is important that you allow enough room for the *highest* number you expect to display.

May I Have That in Writing?

You may have caught the little phrase about printing checks earlier. Printing? On paper? Yes! It is very easy to use the SAY statement to print customized reports, forms and checks. Our old friend, SET FORMAT TO, is used to redirect output from SAY statements to the printer.

SET FORMAT TO, you'll remember from last month, is used to set up an alternate screen entry screen for EDIT and APPEND statements. Its other duty, SET FORMAT TO PRINT, tells dBASE to begin sending the SAY statements to the printer.

When you are using SAY to put things on the screen, you can move all around, putting something in the middle, then at the top, then at the bottom, etc. When you are printing, however, you must start at the top of the page and work down.

The screen has 25 lines on it for SAY statements, but a printer page contains 66. If you had just printed something on Line 5, and then tried to print something on Line 1, it would be printed on Line 1 of the *next* page. dBASE will automatically eject the paper to the top of the page.

We're not going to do any examples of SAY statements today — we'll save that for later. Go ahead and experiment on your own, though. That's the best way to learn.

Project One

Next month we'll begin something different. Instead of going over basic dBASE commands, we'll begin to develop an actual working dBASE II program. The project is a simple one, but one which we can use to learn a lot about dBASE programming technique. We are going to design a mailing list system. It will be a menu driven program that will allow for entry and editing of names, print a mailing list, and print mailing labels. Until then, hit those keyboards!

'On The Air' with the Tandy 200

By Jim Hawk
PCM Contributing Editor

As the news director at a metro Washington, D.C. radio station (Oldies-format WXTR), I've now come to depend on my Tandy 200 (graduating from a Model 100) I'm sure a lot of PCM subscribers do the same: when you head for work in the morning, you don't leave home without it. It's not a certain charge card we're talking about, it's the indispensable 200. After discovering the joys of electronic writing on my M100 while wishing for a bigger screen, I now put a Tandy 200 through its paces every weekday morning, writing and delivering two newscasts an hour. I've found the combination of text processing, communications and non-volatile memory make the 100 or 200 a powerful newsroom tool. And the two machines have given me a competitive edge over the competition by serving as mobile computers in the newsroom, at news conferences, or even going directly on the air reading the script from the LCD screen.

Radio news has the advantage of being absolutely immediate: "This just in . . ." With my trusty portable I can change and revise a newscast virtually right down to the deadline, allowing just enough time for the high-speed dot-matrix printer to churn out the script. TV reporters only get to go on "live" at major events, and usually have to wait till the 6 o'clock news, but radio reporters can use a late-breaking story immediately or in the next hourly newscast at the very least.

On the other hand, radio news means working faster with less: TV stations commonly have 20-30 person news departments, while most radio stations — even in metro markets — have 3 people or less. In many cases it's just one newsperson on duty at any given time. That's why a computer tool like the 200 can help even things up by becoming a go-anywhere word processor, standby wire service and general all-around expedient to getting words down on paper.

And for someone needing ultra-dependable performance, the memory behaves much like the legendary elephant — it (almost) never forgets. I say "almost" because I have crashed the memory twice by accidentally running conflicting machine-language programs. Once you know what to avoid, the Microsoft operating system makes it virtually impossible to lose data other than through a direct KILL command — very handy for 4:30 in the morning before that first cup of coffee.

My first attempt at a newsroom application for the 100 was to create a script and read it directly on the air from the Model 100 screen. Scrolling through the text with the cursor control keys was fun at first, but since radio news is done "live" there's the danger of jumping through stories with one too many cursor commands — "the president said today . . . that the Capitals are 2 and 4 on the season."

Even more disastrous was the occasion when I brought the 100 into the on-air studio. With my electronic script already displayed and just seconds before news time, the low-battery light came on. I found out the hard way just how fast NiCads can die in a Model 100: the red low-battery light came on while I was beginning to read the top line of the screen and it went into auto-shutdown just as I finished that paragraph. By pure luck I had some Associated Press wire copy along as a spare.

It didn't take me too long to realize that while the 100 was useful for the Teleprompter-like application (and the 200 is better with quieter cursor keys and a double-sized screen), I really prefer to hold some genuine paper news copy in my hands when going on the air. That way I can also mark up the script with underlines, slashes and other cues just prior to air time to help in properly phrasing the spoken words.

(Jim Hawk has been working in radio news for the past 13 years and has a science and electronics background. He also does free-lance writing in Washington, D.C.)

The Perfect Newsroom Assistant

My normal routine upon first arriving at the radio station shortly before 5 a.m. is to unpack the 200 and plug it into the extra AC adapter I bought (for six dollars, who's complaining?); then I plug in the printer cable and load the printer with paper so I'm all set to go. How do I know what to write? It's a matter of gathering together wire copy, newspaper articles and our own in-house work including taped audio cuts of sound from the newsmakers called "actualities." Then, I lay it out in semi-disarray and tap out a newscast on the portable computer.

All radio and TV stations depend on the wire services to deliver most of the news — or at least most of the raw material for rewrites. Associated Press dominates the scene with low and high speed printers that spew out national, state and local news, as well as sports and weather. That's all fine as long as the printers, phone lines and fancy black boxes all work at the same time.

But there have been mornings when I've arrived to discover a dead printer (no news!) and once again, the Model 100 or 200 comes to the rescue. I always keep my acoustic cups at work for just such emergencies, so all I have to do is logon to the news areas of CompuServe or Dow Jones Information Service. Instead of having to limp through for hours "faking it" while the wire repairmen arrives, I can keep updated. And maybe in a real emergency, this kind of link could be vitally important.

Satellite and power outages have been rare, but each time, there was something big happening and it didn't seem logical to sit around blind while the portable could get through to the story when there was no other way available.

I enjoy newswriting again now that I've abandoned the old manual Smith-Corona. Once the raw reference material is assembled for writing a newscast, the 200 really shows its stuff. It just so happens that the 200's screen holds a typical story/paragraph. With an average of only thirty or forty seconds to tell any given story, I have to be brief. The 16 line by 40 character screen acts as a good measure of story size — about 100 words — so if I fill up the screen, I'm getting wordy or it's a major top story that needs expanding.

The beauty of preparing a newscast on the 200 (or 100) is the ability to cut and paste with such intuitive ease — even part-timers who work just one day a week had little trouble picking up on the built-in text program. Instead of being a chore with a mechanical typewriter, I've found typing on the 200 not only a breeze, but often downright fun in comparison. Knowing that it takes under two minutes to print, I can keep adding and modifying story details until virtually the last minute.

Sometimes, I'll not realize the time and hear the DJ yell, "You've got 45 seconds before the record's over!" Just about any other computer would be up the proverbial creek without a paddle, but I just calmly unplug from the AC adapter and carry the 200 into the studio with me. Overhead florescent lights cause me to use this procedure only occasionally . . . you have to have the screen at just the right angle to keep the text from being obliterated by the opaque light . . . but much better than trying to make it up off the top of your head!

Printer Tales

During those early notebook portable days — two long years ago in the spring of '83 — not a single battery-powered printer was available for the 100. And with prices for AC-powered units just entering the under $500 range, my station's general manager said, "wait until next year . . . the prices will go down." (Right he was, even though it didn't do me any good then.)

I did wait for the radio station to buy their first computer, and quickly discovered that a tiny line-feed/carriage return program would allow me to use the companion Okidata printer with ease. The only trouble: The printer was in the accounting office about a hundred feet from the newsroom. It got very tiring to write the newscast in one room then carry the computer into the other room to use the printer. But consider how many other computers could do the same thing with such ease . . . and that you can turn on and get the cursor over the desired file while you're walking.

After a few months of this, along came a firm named Axonix Corporation out of Utah with a quick succession of three different battery-powered portable printers. Since I was writing and doing reviews for PCM, I had the good fortune of trying them all out in a real-time, must-work situation. All three held up very well, but I preferred the portable dot-matrix because it was the speediest by far. Since my main enemy is time — typically 15 minutes to write 3½ minutes worth of the news — I like to use as many of those precious minutes as I can actually creating the script, then hit PRINT and get ready for my dash to the studio with audio cartridges, weather, coffee cup and most importantly, the printout in my hands.

There have been the usual disasters such as jammed paper and bad ribbons, but the advantages still far outweigh the disadvantages. When a bulletin or update comes along, the story can simply be modified rather than completely rewritten.

Sold

I don't foresee going back to a typewriter to create news when I've become so accustomed to using the 100 and now the 200 every day. I bought both computer and printer with my own money, while the radio station took a year to investigate and then invested in IBM compatibles. I had to chuckle for almost a year after the "office computers" were installed, though; the database program they bought was so hard to use that the secretary preferred to borrow my 100 to type up *and* print out her mailing lists! I knew then that I'd made the right investment.

Going to PCMfest? Logon to CompuServe's OAG to learn which routing is most economical

Finding out Which Fare is Fair

By Randy Graham
PCM Contributing Editor

OK. Pay attention now. This month you have two assignments to complete. Let me explain: I know you read every word of every issue of PCM, and you remember everything you read. And so, I know that in July as you read this, you still remember the startling announcement publisher Lonnie Falk made in the May issue: He is "considering" having a show for PCM readers in conjunction with the next RAINBOWfest in New Jersey.

I am excited about his announcement because my "other computer" is a Color Computer and I have been going to RAINBOWfests for a couple of years. They are delightfully arranged in an elegant setting — and at the modest costs we Tandy users demand. It is a combination of a family reunion as you meet and hear people you have known in print and can ask them all the questions you want, plus a trade show with an exhibit hall full of products, all of which will run on your machine (and not a single soul will make a disparaging remark about Tandy computers).

Now, your first assignment is to write Lonnie, tell him you think it is a great idea and you want to come. Of course, you will not send him an old plain, vanilla 22¢-stamp letter. Send him a CompuServe EMail (ID #: 70146,1650), or MCI Mail letter ID#: 230-9091). When it takes him a couple of hours connect time to download all those messages, he will be impressed! And, of course, tell him we all want telecommunications to have a prominent place in the show!

Your second assignment is to make plans *now* to attend. It is not too early to be making reservations. Go immediately to the online Official Airline Guide found on several information services and see what flights you can arrange.

You knew all along I was relying on the upcoming PCMfest to lead you into this month's topic, didn't you? You have already guessed this readily available database is the subject for this month's columnn. You may even be in time to make some plans for your vacation. Let's take a look. We will do one of our guided tours where you look over my shoulder as I logon and I will explain what is happening.

The Official Airline Guide started out as a very thick book used by travel agents. This was fine when airlines were regulated and it took years and an act of Congress to change a rate or a fare.

Then they went online with the OAG-EE (Electronic Edition) and decided to make it available to the public through information services like The Source, Dow Jones and CompuServe. Warning: All these services add a surcharge for OAG in addition to their usual fee. This is because they are "gateway" services which means that when you choose "OAG" on the menu, they switch you to OAG's computer and another clock starts running. OAG will bill the service who will add it to your bill. The reason for this approach is the databases are updated very frequently and this is easier to do when you are working with just one central system. 50¢/min $30/hr

OK, let's logon. For demonstration purposes, we will say that I am going from my home in Richmond, Va., to New Brunswick, N.J. on October 18 and return on the 20th. [*Editor's Note: At the time of this writing, PCMfest was scheduled to be held in New Brunswick. It has subsequently been rescheduled to be held in Princeton, N.J., October 11-13. See the PCMfest ad on the inside back cover of this issue.*] On Friday the exhibit hall opens in the evening and program activities and the exhibits run through about 4 p.m. Sunday. I would like to catch a flight after work Friday and return Sunday evening. Now, when I logon and get to OAG, this is what happens.

```
WELCOME TO THE OFFICIAL AIRLINE GUIDE

(OAG), COPYRIGHT 1985, OFFICIAL AIRLINE
GUIDES, INC., OAK BROOK, ILLINOIS 60521

PRESS RETURN FOR SUBSCRIBER BULLETIN
OR ENTER /F, /S, /I, /U
ENTER /M FOR A LIST OF OAG EE COMMANDS
```

(Randy Graham is a rehabilitation counselor working with the handicapped. Personal computing is his hobby; telecommunications, one of his favorite activities. He has done free-lance information retrieval and is an inveterate user of the major online systems.)

These slash commands give me the option of requesting fare or schedule information and information or feedback to the database. Let's look at schedules first:

```
     /S

ENTER DEPARTURE CITY NAME OR CODE
     RICHMOND,VA

ENTER DESTINATION CITY NAME OR CODE
     NEW BRUNSWICK,NJ

YOUR DESTINATION CITY DOES NOT HAVE AIR SERVICE.
BELOW IS A LIST OF CITIES/AIRPORTS SERVING-NEW
BRUNSWICK,NJ,USA
1         NYC-NEW YORK      -ALL AIRPORTS
2  22 MI  EWR-NEW YORK,NY,USA/NEWARK
3  37 MI  LGA-NEW YORK,NY,USA/LA GUARDIA
4  47 MI  JFK-NEW YORK,NY,USA/KENNEDY

ENTER LINE NUMBER OR ENTER ANOTHER CITY NAME OR
CODE.
```

This is the first useful data. Nothing but a helicopter can land in New Brunswick; I will have to fly into another airport. Newark will be most convenient, so we select that.

```
     2

ENTER DEPARTURE DATE OR PRESS RETURN TO USE 26 MAY
     18 OCT

ENTER DEPARTURE TIME OR PRESS RETURN TO USE 600AM

          DIRECT FLIGHTS      FRI-18 OCT
FROM-RICHMOND,VA,USA
# TO-NEW YORK,NY,USA/NEWARK
1  920A  RIC 1040A  EWR AA1412 SWM   0
2 1115A  RIC  110P  EWR AA1404 SWM   1
3  110P  RIC  210P  EWR PI 774 F28 S 0
4  300P  RIC  500P  EWR AA1446 SWM   1
5  500P  RIC  600P  EWR PI 776 F28 S 0
6  635P  RIC  755P  EWR AA1456 SWM   0
ENTER +,-,CX,X#,F#,RS    (#=LINE NUMBER)
```

OK, there is a scheduled flight leaving at 6:35 which arrives in Newark at 7:55. Let's check return flights.

```
     RS

ENTER RETURN DATE OR PRESS RETURN KEY TO USE 18
OCT
     20 OCT

ENTER RETURN TIME OR PRESS RETURN KEY TO USE 600PM
     400PM

          DIRECT FLIGHTS      SUN-20 OCT
FROM-NEW YORK,NY,USA/NEWARK
# TO-RICHMOND,VA,USA
 NO EARLIER DIRECT FLIGHT SERVICE
1  850A  EWR  952A  RIC PI 769 F28 S 0
2  200P  EWR  355P  RIC PI  55 72S   1
3  305P  EWR  407P  RIC PI 777 F28   0
4  530P  EWR  730P  RIC AA1449 SWM   1
5  655P  EWR  757P  RIC PI 779 F28 S 0
6 1010P  EWR 1110P  RIC PI 489 73S   0
 NO LATER DIRECT FLIGHT SERVICE
PRESS RETURN KEY FOR CONNECTIONS
ENTER CX,X#,F#,RS        (#=LINE NUMBER)
```

That 5:30 flight looks pretty good; let's check fares.

```
     /F

ENTER DEPARTURE CITY NAME OR CODE OR PRESS RETURN
TO USE NEW YORK,NY,USA/
     RIC

ENTER DESTINATION CITY NAME OR CODE OR PRESS
RETURN TO USE RICHMOND,VA,USA
     EWR

ENTER DEPARTURE DATE OR PRESS RETURN TO USE 18 OCT
OR ENTER 99 FOR FARES ON ALL DATES

               FARE MENU
FARES FOR DIRECT FLIGHTS          1  COACH AND
BUSINESS CLASS FARES        6
2  FIRST CLASS AND EQUIVALENT FARES    7
3  COACH, BUSINESS AND FIRST CLASS     8
4  ADVANCE-PURCH AND EXCURSION FARES   9
5       ALL OF THE ABOVE FARES         10

PLEASE ENTER A NUMBER
```

"Supersaver" fares are usually the best buy; let's investigate them.

```
     4

     TRIP TYPE MENU
1 ONE-WAY
2 ROUND-TRIP - RETURN DATE UNKNOWN
3 ROUND-TRIP

PLEASE ENTER A NUMBER
     3

ENTER RETURN DATE OR PRESS RETURN KEY TO USE 18
OCT
     20 OCT

FARES IN US DOLLARS              FRI-18 OCT
SELECTED FOR RIC-EWR

#  ONE-WAY  RND-TRP ARLN/CLASS FARECODE
 NO LOWER FARES IN CATEGORY
1*            113.00   PI/W     WE30X23
2*            113.00   PI/W     WE30Z23
3*            113.00   AA/Q     QE30
```

```
4*           113.00 PI/W-PI/W  WE30X23
5*           113.00 AL/Q-AL/Q  QE30
6*           113.00 PI/W-PI/W  WE30Z23
7*           172.00   PI/B     BE70
8*           172.00 PI/B-PI/B  BE70
 * ENTER L# TO VIEW LIMITATIONS
ENTER +,L#,X#,S#,R#,M,RF(#=LINE NUMBER)
```

See all those little codes about special conditions? Let's get some more information about them.

```
     L1

LIMITATIONS DISPLAY            FRI-18 OCT
RIC-NYC     CLASS:W  FARECODE:WE30X23
PIEDMONT AVIATION
FARE DESCRIPTION: EXCURSION FARES
BOOKING CODE: W.
FARE IS ONLY AVAILABLE FOR TRAVEL ON  THU THRU
MON.  FARE IS NOT AVAILABLE FOR TRAVEL FROM  22
NOV 85 THRU 23 NOV 85; AND FROM 26 NOV 85 THRU 27
NOV  85.  FARE IS NOT AVAILABLE FOR TRAVEL FROM 01
DEC 85 THRU 02 DEC 85; AND FROM 20 DEC 85 THRU 22
DEC 85; AND FROM 27 DEC 85 THRU 29 DEC 85; AND
FROM 01 JAN 86 THRU 02 JAN 86.  FARE IS NOT
AVAILABLE FOR TRAVEL ON 05 JAN 86.  TRAVEL MUST
BEGIN ON OR AFTER 24 MAY 85.
MINIMUM STAY IS OVERNIGHT ON SATURDAY.
MAXIMUM STAY ALLOWED IS 21 DAYS.
RESERVE TICKET FOR TRAVEL NO LATER THAN 30 DAYS
BEFORE DEPARTURE.
PURCHASE TICKET FOR TRAVEL NO LATER THAN 30 DAYS
BEFORE DEPARTURE, BUT NO LATER THAN 14 DAYS AFTER
TIME OF RESERVATION.
THE PENALTY FOR CANCELLING YOUR TRAVEL IS 25
PERCENT OF THE FARE.
 * END OF LIMITATIONS DISPLAY *
ENTER S TO VIEW SELECTED PI SCHEDULES
ENTER F TO RETURN TO FARES DISPLAY
     /Q
```

```
END OF OAG SESSION
```

The special fares will apply for the dates I want, but I must get my ticket 30 days in advance and I will pay a hefty penalty if I cancel. Now let's try it for *your* hometown.

There are a couple of problems about using OAG online. One is certainly the cost. That search only took me about five minutes, but remember, it costs a lot more than CompuServe to access.

Related to that consideration is the fact that any travel agent will do the same search for free in hopes of getting your business; they consider it a normal operating expense. I cannot see why you would use it unless you just like to do your own planning and exploring online.

Another problem is you cannot make reservations on OAG. You have to go to an agent or the airline for that. The services which have a gateway to OAG also offer online travel services which will help you with reservations and ticket purchases.

A final small point is OAG has its own set of commands to learn and it is not very user friendly. I deleted from our printout several instances where a typographical error made me read a pretty long error message and reprinted menu. When I am paying 50¢ a minute, I only want to read those things once!

Now you are on your way to the first-ever PCMfest. See you there. One final note: As I was writing this, the June issue arrived and the aforementioned Lonnie Falk tells us he has joined —or been forced into — citizenship in our "Electronic Village." Welcome to the club, Lonnie! PCM

THE TANDY 200 AND MODEL
OFFER MUCH MORE THAN J

Our accessories make them the only co

When we introduced the Advanced Technology Model 100, it quickly became the most highly-acclaimed lap-size computer ever. A battery or AC-powered computer with five built-in programs and a telephone modem that fits into a briefcase. Our Tandy 200 took this great idea a few steps further—a big easy-to-read 16-line by 40-character "pop-up" display, Multiplan spreadsheet software, plus an alarm and calculator.

But the story doesn't end there—we back both machines with extensive accessories that help make them powerful desktop tools.

New Low Price! Disk/Video Interface

Turn your Tandy 200 or Model 100 into a versatile disk-based home or office system with a Disk/Video Interface—now only $499 (26-3806, was $799.00 in Cat. RSC-14). It features a built-in 184K floppy disk drive, and a 40-character by 25-line display with your TV or an 80 × 25 display with a standard video monitor. You can even add a second internal disk drive.

Thermal Ribbon Printer

Use it anywhere! The TRP-100 (26-1275, $299.95) works on batteries or AC power (adapter included), and prints on plain or thermal paper. The TRP-100 quietly prints 50 characters per second. Produces bit-image graphics, too.

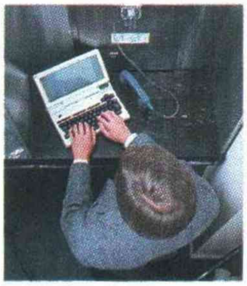
Versatile Bar Code Reader

Our Bar Code Reader (26-1183, $99.95) is an optical scanning device that's ideal for inventory control, billing and couponing. Software drivers, callable from BASIC, allow you to read Universal Product Code, 3 of 9, Plessey and other bar code formats.

100 PORTABLE COMPUTERS
ST COMPACT PORTABILITY

puter you'll need—in or out of the office.

Communications Power



One of the most exciting features of the Tandy 200 and Model 100 is the built-in modem that lets you communicate with other computers over the phone line. Our Direct-Connect Modem Cable comes with one free hour on both the CompuServe Information Network and Dow Jones News/Retrieval service.

Get news, weather and sports, business and stock market information and more. Information networks open up a whole new world of applications power.

And when you're on the road, you can bypass direct connection with our Acoustic Coupler. It's a great way to keep in touch with your home office. And we even offer a briefcase and padded carrying case to make portability all it's supposed to be.

Come in Today!

Stop by your local Radio Shack Computer Center and see both the Advanced Technology 24K Model 100 (26-3802, $599) and 24K Tandy 200 (26-3860, $999), as well as our complete line of accessories and software. You'll find everything you need for professional, portable computing power—for your desk or on the road.

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

Prices apply at Radio Shack Computer Centers and participating stores and dealers. Multiplan/TM Microsoft Corp. CompuServe/Registered TM CompuServe, Inc. Dow Jones News/Retrieval/Registered TM Dow Jones & Co., Inc.

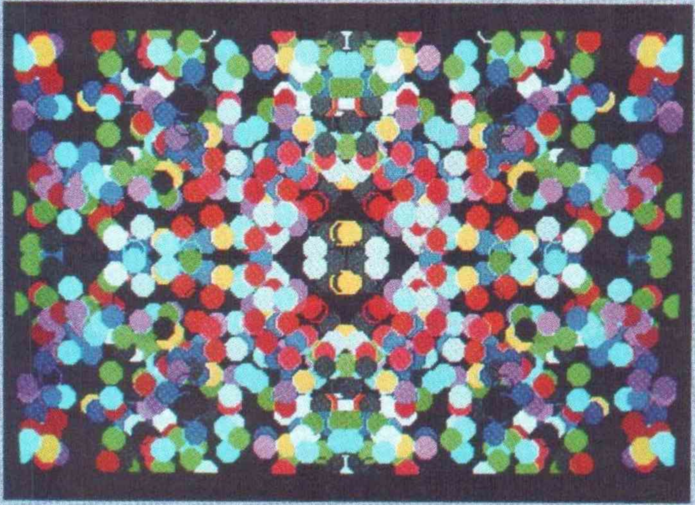
Send Me a 1985 Computer Catalog!

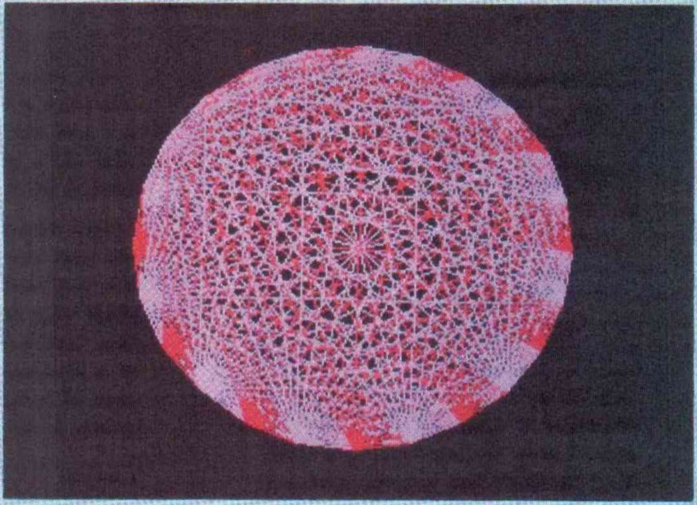
Mail To: Radio Shack, Dept. 86-A-86, 300 One Tandy Center, Fort Worth, Texas 76102

NAME ______
ADDRESS ______
CITY ______
STATE ______ ZIP ______
PHONE ______

The Gallery

By Wayne Sanders





You're off doing something else. Your computer patiently awaits your return to its keyboard. A lonely A> prompt shares the vastness of the screen only with the blinking cursor. This month's Gallery programs bring life to the void with colorful random graphics.

The first program, KSCOPE.BAS, draws a kaleidoscopic image of randomly-colored circles on the screen. The second, WEBS.BAS, creates multicolored "spider-webs" by attaching points on a circle. Each turns your temporarily-inactive computer screen into an aesthetically pleasing piece of art in motion.

Both programs may be run on either a Tandy 1000 or 2000. If you are running them on a Tandy 2000, change the first line of each program to read MODEL=2000.

Be a Gallery Guest Artist

The Gallery is always looking for new art. Send us your masterpiece on disk in the form of a BASIC program or a loadable .BIN file. A color screen dump would also be most helpful. It need not be a BASIC program, but if it is, we would prefer versions that would work on all Tandy MS-DOS machines. If your work is selected for exhibit, you will be awarded a $25 Gallery prize. Good Luck!

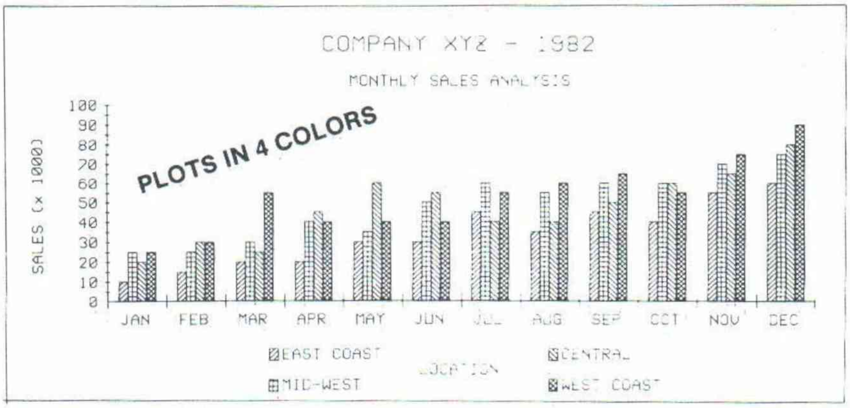
Listing 1: KSCOPE.BAS

```
100 MODEL=1000
110 IF MODEL=1000 THEN CLEAR ,,,32768!:SM=5:XR=320:YR=200:CR=15:SR=50
120 IF MODEL=2000 THEN SM=3:XR=640:YR=400:CR=7:SR=100
130 SCREEN SM:KEY ON:CLS:KEY OFF
140 R=VAL(MID$(TIME$,7,8))*VAL(MID$(TIME$,4,2)):RANDOMIZE R
150 PNTS=INT(RND*25)+5:SIZE=INT(RND*SR)+SR
160 C1=INT(RND*CR)+1:C2=INT(RND*CR)+1:IF INT(RND*2)=1 THEN C2=C1
170 FOR J=0 TO PNTS-1
```

```
180     I=6.26/PNTS*J
190     X=SIN(I)*SIZE+XR/2:Y=COS(I)*SIZE+YR/2
200     FOR K=0 TO PNTS-1
210        I=6.26/PNTS*K
220        X1=SIN(I)*SIZE+XR/2:Y1=COS(I)*SIZE+YR/2
230        IF K MOD 2=0 THEN C=C1 ELSE C=C2
240        LINE (X,Y)-(X1,Y1),C
250     NEXT K
260 NEXT J
270 FOR I=1 TO 1000:NEXT I
280 IF INT(RND*2)=1 THEN CLS
290 GOTO 150
```

Listing 2: WEBS.BAS

```
100 MODEL=1000
110 IF MODEL=1000 THEN CLEAR ,,,32768!:XR=160:YR=100:CR=16:CS=5:SM=5
120 IF MODEL=2000 THEN XR=320:YR=200:CR=8:CS=10:SM=3
130 SCREEN SM:KEY ON:CLS:KEY OFF
140 X=INT(RND*XR):Y=INT(RND*YR):C=INT(RND*CR)
150 CIRCLE (X,Y),CS,C:PAINT (X,Y),C,C
160 CIRCLE (XR*2-X,Y),CS,C:PAINT (XR*2-X,Y),C,C
170 CIRCLE (X,YR*2-Y),CS,C:PAINT (X,YR*2-Y),C,C
180 CIRCLE (XR*2-X,YR*2-Y),CS,C:PAINT (XR*2-X,YR*2-Y),C,C
190 GOTO 140
```



The Portable Machine

By Robert D. Covington

Could you imagine your Model 100 without a keyboard? Just imagine turning on the computer, looking at the nice screen and being able to do nothing about it (except turn off the computer). The keyboard is the Model 100's main input device. Just as the LCD is the 100's main output device to the "real world," the keyboard is the "real world" main input device.

Because of the importance of the keyboard, it is necessary that you learn to effectively use the keyboard in your machine language programs.

Keyboard Hardware

The Model 100's keyboard is organized in a 9-by-8 matrix as shown in Table 1. When the computer checks for a pressed key, a false pulse (0) is sent through one column while a true pulse (1) is sent through all other columns; then the computer checks each row for the presence of a false pulse.

If one is present, the computer assumes the key at the coordinates of the false row (0) and the selected false column (0) is pressed. If a false pulse is not present, the computer assumes no keys are pressed in that column and repeats the whole process for the next column. Once all the columns and rows have been checked, the computer can then assume no key has been pressed.

To help explain this, let's see how the computer would detect the 'F' key being pressed. The computer would start by sending a 11111110 out port 0B1H and a 1 out port 0B2H to deselect all but the first column. Following this, the computer would read any of the ports from 0E0H through 0EFH to load in the 8-bit row status. Since no keys in the first column are pressed, the port status should be 11111111. With no success in the first column, the computer proceeds to scan the second column by sending a binary 11111101 out port 0B1H and a 1 out port 0B2H.

If you look at the matrix in Table 1, this column contains the A, S, D, F, G, H, J and K keys. Since the 'F' key is being pressed, the row status should read a binary 11101111 (bit 3 reset). Once the computer has found a key, it then scans the last two columns for the SHIFT, CONTROL, GRPH, CODE or CAPS LOCK keys being pressed. Then, by using the table starting at 7BF1H, the computer converts the key sequences into actual ASCII characters.

ROM Routines

This process of scanning the keyboard can be quite cumbersome to perform in machine language. Like always, the Model 100 ROM comes to the rescue by providing a set of machine language subroutines for accessing the keyboard.

In most cases, when the keyboard is scanned by machine language programs you are usually only interested in the ASCII value of the key being pressed. In addition, it is usually very important that the computer detect every key pressed even if the routine accessing the keyboard is slower than the typist. To achieve these two goals, the Model 100 ROM scans the keyboard as a background interrupt task.

If you recall back in the March 1985 installment of this series, I described an interrupt that performed an RST 7.5 (a call to 003CH) approximately every four microseconds. This interrupt is used by the Model 100 primarily to scan the keyboard for possible characters. If a key is found by the interrupt routine to be pressed, the computer converts the key into its ASCII equivalent and stores it in a temporary buffer called the "keyboard queue."

As more keys are pressed, the keyboard queue grows until it reaches its maximum length of 32 characters. On the Model 100, this keyboard queue starts at 0FFABH and extends to 0FFCBH. In addition, the byte at 0FFAAH specifies how many characters are currently in the queue.

Since the keyboard queue is filled as a background task, it can become quite difficult for a machine language program to monitor the queue's current status. To help with this, the Model 100 ROM contains six major routines for checking the queue and the keys in it.

The routines at 13DBH and 7270H check for the presence of pending characters in the keyboard queue. If one or more characters are pending in the keyboard queue, these routines set the 'Z' flag. In addition, the routine at 7270H sets the 'C' flag if a break (CONTROL C) is being held in the queue.

The routine at 5F2FH empties the keyboard queue till a space is found. If no space is found in the queue, the routine waits till one is entered from the keyboard. The routine at 7283H checks for a break (CONTROL C) or pause (CONTROL S). If one of those control sequences is found, the routine returns with the 'C' flag set and the 'Z' flag reset.

(Bob Covington has been a computer programmer and consultant for the past five years, most recently focusing his attention on both the Model 100 and the 2000. He is also a technical writer and editor who has just completed the technical editing on an upcoming book on Model 100 machine language.)

To retrieve a character from the keyboard queue, the routine at 7242H can be used. This routine takes the character that has been in the buffer the longest, loads it into Register A and deletes it from the buffer. If the character retrieved is a function key, LABEL, PRINT, SHIFT-PRINT, or PASTE, the 'C' flag is set and Register A contains a number from 0-11 describing the key pressed (see 7242H description in Table 2 for more detail). If no characters are present in the type-ahead buffer, the routine returns with 'A' equaling zero and the 'Z' flag set.

To help understand how this routine works, type in, assemble and execute Program 1. This program prints the character code returned in Register A on the screen. If an asterisk precedes the number, then a special character has been pressed.

If you don't want to detect function keys but want their programmed ASCII equivalent, the routines at 12CBH and 5D64H can be used. These routines, like 7242H, load Register A with the next character in the type-ahead buffer. Unlike 7242H, if a function key is pressed, these routines automatically dump the programmed function key text into the type-ahead buffer. This, in effect, allows preprogrammed function key text to be entered into a machine language program just as if it were typed on the keyboard.

The only differences between the routine at 12CBH and 5D64H is the routine at 5D64H converts all alpha characters to uppercase prior to returning with the character. With both routines, the computer waits for a key to be pressed before returning back to the calling program.

So far, I have discussed routines which just input single characters from the keyboard. If you need to enter an entire text string from the keyboard, the routines at 463EH and 4644H can be used. These routines access BASIC's INPUT routine and load the string input from the keyboard into the memory starting at 0F685H. Like BASIC's INPUT, up to 255 characters can be entered and edited with all characters going to the screen.

The only differences between the routine at 463EH and 4644H is the routine at 463EH prints a question mark prior to accepting keyboard input. Upon exit from these subroutines, HL is equal to 0F685H (where the text starts) and the 'C' flag is set if a CONTROL C or SHIFT-BREAK was pressed. Under normal circumstances,

Program 1
7242H Demo Routine

```
;*** Keyboard Scan Demo
;*** By Robert D. Covington
;
        ORG     0FCC0H
START    CALL    7242H       ;SCAN KEYBOARD
        JP      Z,START     ;LOOP IF NO CHR
        JP      NC,SKIP     ;JP IF NORM CHR
        PUSH    AF          ;SAVE CHARACTER
        LD      A,42        ;42="*"
        RST     20H         ;PRINT *
        POP     AF          ;RESTORE A
SKIP     LD      L,A         ;PUT CHAR IN L
        LD      H,0         ;ZERO MSB
        CALL    39D4H       ;PRINT #
        CALL    4222H       ;PRINT CRLF
        JP      START       ;LOOP
        END     START
```

Table 1
Model 100 Keyboard Matrix

	7 ¦	L	K	I	?	*	->	Ent	f8	Brk
Row	6 ¦	M	J	U	>	&	<-	Prt	f7	
In	5 ¦	N	H	Y	<	^	Up	Lbl	f6	Cap
	4 ¦	B	G	T	"	%	Dwn	Pst	f5	Num
0E0H-	3 ¦	V	F	R	:	$	+	Esc	f4	Cde
0EFH	2 ¦	C	D	E	]	#	−	Tab	f3	Gph
(224-	1 ¦	X	S	W	P	@	)	Del	f2	Ctl
239)	0 ¦	Z	A	Q	O	!	(	Spc	f1	Sft
	bit	0	1	2	3	4	5	6	7	0

```
        [----------- 0B1H or 0B9H ------------] [0B2H]
        [----------- 177  or 185 -------------] [178 ]
                    Column Strobe
```

Note: This table is incorrectly documented in Radio Shack's *Model 100 Technical Reference Manual* (26-3810).

Table 2
Model 100 Keyboard Memory Map

```
003CH -  8085 RST 7.5 interrupt vector.  This interrupt is
        generated by the internal timer at regular 4
        microsecond intervals.  On each interrupt, the Model
        100 performs a keyboard scan and updates the type ahead
        buffer if nescessary.  The RAM vector F5FFH is called
        on each interrupt.

0CA3H -  INPUT Statement

12CBH -  Wait for a key from keyboard.  All function keys are
        automatically converted to their programmed text.
         Exit:
                A - Character entered
           C flag - Set if special character

13DBH -  Check keyboard queue for pending characters
         Exit:
                A - Destroyed
           Z flag - Set if no characters pending

1AB2H -  KEY() statement
```

```
1AC3H -  KEY STOP/ON/OFF statements

1B32H -  RST 7.5 interrupt routine (see 3CH).

1BB8H -  KEY Statement

1BBDH -  KEY LIST Statement

4009H -  Clear all COM, TIME, and KEY interrupt definitions
         Exit:
               HL, A, and B destroyed

463EH -  Input line and place at F685H.  Print characters as
        they are inputed.  Start input with a "? ".
         Exit:
            F685H - Text input from keyboard
               HL - Loaded with F685H
           C flag - Set if input aborted by a control C

4644H -  Same as 463EH but no "? " is printed.

4684H -  Input routine 463EH Control C handler

4696H -  Input routine 463EH ENTER handler

46A0H -  Input routine 463EH backspace, left arrow, control H
        handler

46C3H -  Input routine 463EH Control U & X handler

46C3H -  Input routine 463EH Control U & X handler

4BEAH -  INKEY$ Function

5A79H -  Clear function key definition table
         Exit:
               All registers destroyed
```

the routine should return with the last character entered being the ENTER key.

Both of the above routines (12CBH and 463EH) support dumping the text associated with each of the function keys into the keyboard queue. Under most circumstances, the programmed text associated with each function key is stored in the 128 byte area starting at 0F789H. This table holds up to 16 characters for each of the eight function keys (16*8=128). Any characters that are not filled with programmed text are padded (filled) with nulls (0).

To modify the definition of text associated with a function key under machine language control, two options exist. The first option requires that the machine language program directly load a new definition for one or all of the function keys on top of the old definition(s). The second option is to use the routine at 5A7CH to completely load in all new function key definitions.

To define a function key using this routine, HL needs to point to an in-memory table containing all of the text associated with each function key. Unlike the format of the actual function key definition area at F789H, this subroutine requires that the last character of each function key definition

```
5A7CH -  Set new function key table.  The table contains the
        function key definitions up to 16 characters for each
        of the 8 function keys.  The last byte of each entry
        should have the high bit set to signify the end of the
        function key definition.  Empty entries should contain
        an 8ØH.
         Entry:
               HL - Points to function key table
         Exit:
               All registers destroyed

5D64H -  Wait for character from keyboard and convert it to
        uppercase if necessary.  All function keys are
        automatically converted to their programmed text.
         Exit:
                A - Uppercase character from keyboard

5F2FH -  Wait for a space to be entered on keyboard
         Exit:
                A - Space (2ØH)

7242H -  Scan keyboard for a possible character (Note:
        Shift-Break is converted to a Control C (Ø3))
         Exit:
                A - Character from keyboard
           Z flag - Set if no key is found
           C flag - Set if special function key
                     A          Key
                 -----------------------
                   Ø-7          F1-F8
                     8          LABEL
                     9          PRINT
                   ØAH          SHIFT-PRINT
                   ØBH          PASTE

727ØH -  Check keyboard queue for pending characters
         Exit:
                A - Destroyed
           Z flag - Set if charcter pending
           C flag - Set if break is present

7283H -  Check for break (Control C) or wait (Control S)
         Exit:
                A - Destroyed
           C flag - Set if Control C or S
           Z flag - Set if no control C or S

729FH -  Check to see if SHIFT BREAK is being pressed.
         Exit:
                A - Destroyed

           Z flag - Set if shift break is being pressed
           C flag - Set if shift break is being pressed

72B1H -  Scan BREAK, CAPS, NUM, CODE, GRAPH, CONTROL, and SHIFT
        key column of keyboard.  Reset the bit in the A
        register that corresponds to the key being pressed
         Exit:
```

have its high bit set. In the case of a null entry, an 80H (a null with the high bit set) should be used. This routine does not require the programmer to pad the text with nulls.

An example of a function key definition table for the routine at 5A7CH is:

```
DEFM    Fil
DEFB    229     ;'e' OR 80H — F1
DEFM    Loa
DEFB    228     ;'d' OR 80H — F2
DEFM    Sav
DEFB    229     ;'e' OR 80H — F3
DEFM    Ru
DEFB    238     ;'n' OR 80H — F4
DEFM    Lis
DEFB    244     ;'t' OR 80H — F5
DEFB    80H     ;Null entry — F6
DEFB    80H     ;Null entry — F7
DEFM    Men
DEFB    245     ;'u' OR 80H — F8
```

If you wish to clear (erase) the function key definitions, the routine at 5A79H loads a null function key definition over the current definitions. By clearing this table, the programmer doesn't have to worry about text spilling into an input routine from the user pressing a function key.

This is especially nice when you program menus that require only one key to be pressed to select an option. Wouldn't it be nice if F1 was defined to DRY and the keys 'D,' 'R' and 'Y' deleted all the files used by your program? As a general rule, always clear the function key definitions unless your machine language program properly supports them.

Most of the ROM routines discussed thus far use the interrupt supported keyboard queue. In some instances, however, interrupts need to be disabled and the keyboard queue cannot be used.

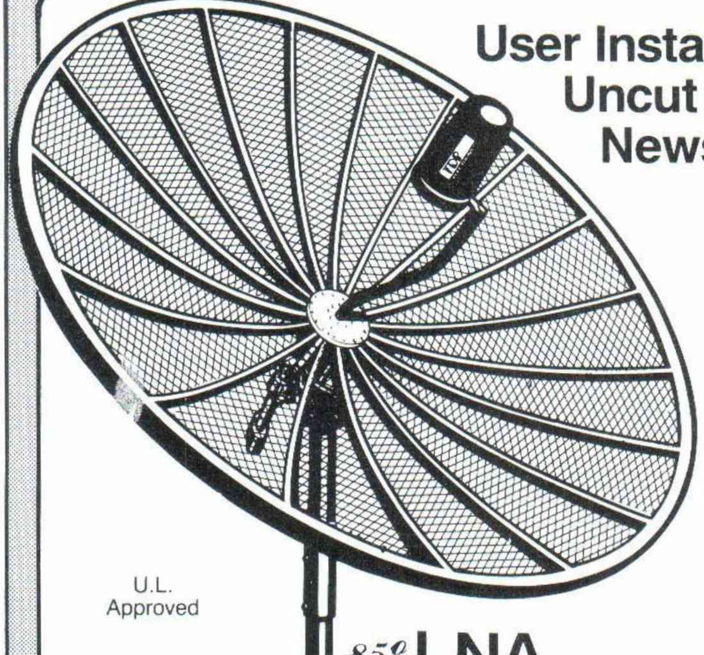
A prime example of this is high-speed

Factory Authorized

Radio Shack® DEALER

Discover a Whole New World of TV Entertainment!

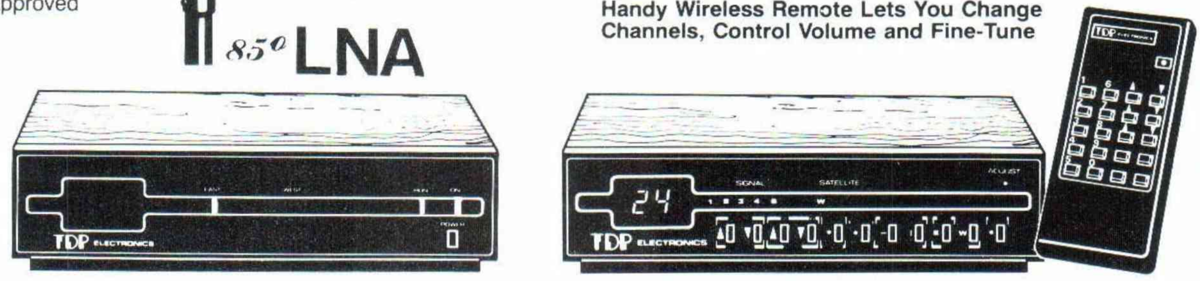
SATELLITE SYSTEM



User Installable! Over 100 Channels! Uncut Movies! Specials! Sports! News! Weather! Music Videos!

Entire System **$1999**

- Totally Automatic System With 10½-Ft. Dish, Receiver and Wireless Remote Control
- Comes With All Accessories Needed for Complete Do-it-Yourself Installation
- Descrambler Ready



Handy Wireless Remote Lets You Change Channels, Control Volume and Fine-Tune

Now you can enjoy it all! 24-hour non-stop entertainment that you simply can't get from local TV, cable, and even video tape! You'll see the hottest stars, recent uncut and uninterrupted movies, the ultimate in sports coverage—and the best part is, there are no monthly fees to pay. You'll see exciting family-oriented programs, plus music videos, FM broadcasts and movies with provision for stereo, and 24-hour news coverage on both TV and radio. It all adds up to out-of-this-world TV viewing at a down-to-earth price. Come in today and discover how easy it is to enjoy satellite programming! Includes complete installation manual.

(817) 825-4027
NOCONA ELECTRONICS • Box 593 • Nocona, TX 76255

UPS Shipping

A RADIO SHACK DEALER

digital signal processing like cassette I/O. During cassette I/O, the CPU cannot have its time-critical routines interrupted for anything except a power down. Under these circumstances, the interrupts that service the keyboard queue have to be disabled and normal keyboard I/O is brought to a halt.

As you know, however, all cassette I/O is capable of being halted by pressing SHIFT-BREAK. To do this scan, the ROM contains two subroutines for finding the status of the last column of the keyboard matrix. The routine at 72B1H loads Register A with the row status of the last column. Just like with a normal keyboard scan, any reset bits reflect a key being pressed.

In the case of scanning for a SHIFT-BREAK, Register A would contain a 126 (01111110 binary). If the only sequence you are interested in is SHIFT-BREAK, the routine at 729FH can be used. This routine sets the 'C' and 'Z' flags if the SHIFT-BREAK keys are pressed. Both of these routines are quite handy in providing keyboard exits to time-sensitive machine language routines.

Model 100 Assembler Notes

It seems the Model 100 assembler published in the April 1985 issue (Page 31) of PCM has a few bugs in it. When I went through to fix the bugs, I discovered a few ways to optimize the code and make the assembler slightly smaller and faster.

Table 3 contains a list of all the lines that need to be changed in the original assembler. Any line with no special instructions should be entered as shown in the table. Any line preceded with an asterisk should only be entered if the program is to be run on a Model 1000/1200/2000.

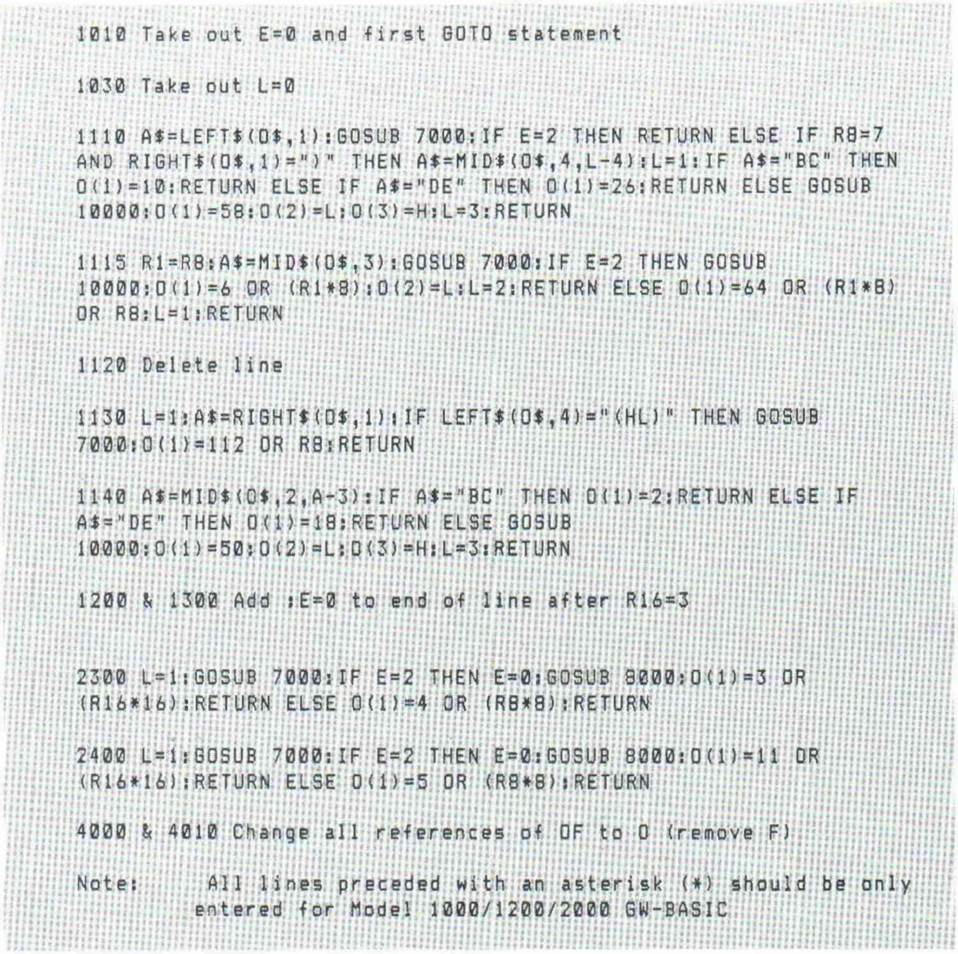
One important note I forgot to mention in the April issue was that if you plan to load a machine language program in memory anywhere under 0F5F0H, BASIC's high memory pointer needs to be set. In BASIC, the area allocated for variable space is always located in the upper regions of memory. If a BASIC program is used to poke in the machine language program, the program could overwrite the variable table and problems could result.

To alleviate this problem, you need to tell BASIC what area of high memory it shouldn't use. In most cases, you will want to set the high memory pointer with the value of the smallest ORG assembler instruction. To set high memory, use the section option of BASIC's CLEAR instruction (i.e., CLEAR

```
               A - Each reset bit coresponds to a key being
                   pressed (BREAK=bit 7, CAPS=bit 5, etc.)

7BF1H -  Start of keyboard conversion matrix.

7D32H -  End of keyboard conversion matrix.

F5FFH -  RST 7.5 RAM vector (3)  Used for the timer interrupt
        (Called)

F685H -  Start of keyboard buffer used by the keyboard line
        input routine at 4644H.

F789H -  Function key definition area (128)

F80AH -  Function key definition area used by BASIC (128)

F94AH -  Function key vector table (24)
            First byte - On/off status
           Next 2 bytes - Vector address

F9A5H -  Start of Paste buffer (Hayashi) (2)

FB0CH -  TERM F6 RAM vector (2)

FB0EH -  TERM F7 RAM vector (2)

FF97H -  8 bits for storing space, del, tab, esc, paste, label,
        print, and enter key recognition.

FF98H -  8 bits for storing function key recognition

FFA2H -  8 bits for storing shift, ctrl, grph, code, num, caps
        lock, and break key (last column).

FFAAH -  Number of characters in keyboard buffer.

FFABH -  Start of keyboard typeahead buffer (32)
```

Table 3
Model 100 Assembler Additions/Corrections plus changes for the 1000/1200/2000

```
100 Change all references of FN$ to F$

*100 & 930 Change MENU instructions to END or SYSTEM

115 Replace PL=PL-1 with PS=PL

*120 Change LN$="LCD:" to LN$="SCRN:"

*125 CLS:LOCATE 12,35:PRINT"Compiling"

130 EC=0:PC=0:L=0:DO$="":PL=PS:OPEN F$ FOR INPUT AS 1:OPEN LN$
FOR OUTPUT AS 2:IF NOT OF THEN OPEN OF$ FOR OUTPUT AS 3

160 & 300 Remove Beep Statments if desired

300 BEEP:IF OF THEN 150 ELSE FOR X=1 TO
L:DO$=DO$+STR$(O(X))+",":NEXT

310 IF LEN(DO$)>200 OR EP>-1 THEN
PRINT#3,PL;"DATA";LEFT$(DO$,LEN(DO$)-1):PL=PL+1:DO$=""

320 IF EP<0 THEN 150

330 & 340 Delete Lines

500 PRINT #3,PL;"FORPP=";S;"TO";EP-1;":READPX:POKE
PP,PX:NEXT:CALL";ES:PL=PL+1:GOTO150

1000 L=0:E=0:EP=-1:IF LEFT$(A$,1)=";" OR A$="" THEN RETURN
```



```
1010 Take out E=0 and first GOTO statement

1030 Take out L=0

1110 A$=LEFT$(O$,1):GOSUB 7000:IF E=2 THEN RETURN ELSE IF R8=7
AND RIGHT$(O$,1)=")" THEN A$=MID$(O$,4,L-4):L=1:IF A$="BC" THEN
O(1)=10:RETURN ELSE IF A$="DE" THEN O(1)=26:RETURN ELSE GOSUB
10000:O(1)=58:O(2)=L:O(3)=H:L=3:RETURN

1115 R1=R8:A$=MID$(O$,3):GOSUB 7000:IF E=2 THEN GOSUB
10000:O(1)=6 OR (R1*8):O(2)=L:L=2:RETURN ELSE O(1)=64 OR (R1*8)
OR R8:L=1:RETURN

1120 Delete line

1130 L=1:A$=RIGHT$(O$,1):IF LEFT$(O$,4)="(HL)" THEN GOSUB
7000:O(1)=112 OR R8:RETURN

1140 A$=MID$(O$,2,A-3):IF A$="BC" THEN O(1)=2:RETURN ELSE IF
A$="DE" THEN O(1)=18:RETURN ELSE GOSUB
10000:O(1)=50:O(2)=L:O(3)=H:L=3:RETURN

1200 & 1300 Add :E=0 to end of line after R16=3


2300 L=1:GOSUB 7000:IF E=2 THEN E=0:GOSUB 8000:O(1)=3 OR
(R16*16):RETURN ELSE O(1)=4 OR (R8*8):RETURN

2400 L=1:GOSUB 7000:IF E=2 THEN E=0:GOSUB 8000:O(1)=11 OR
(R16*16):RETURN ELSE O(1)=5 OR (R8*8):RETURN

4000 & 4010 Change all references of OF to O (remove F)

Note:     All lines preceded with an asterisk (*) should be only
        entered for Model 1000/1200/2000 GW-BASIC
```

100,60000 where 60000 is the high memory pointer).

Another note I failed to mention was that "Label not found" errors can be erroneously generated on the first pass. If the assembler stops after the first pass because of an "Illegal Opcode" error, the label errors should be ignored. The reason these label errors are generated is because the labels might not all be defined until the second pass. In most cases, "Label not found" errors are not significant unless the assembler has gone through two entire passes.

Conclusion

In Table 2, you will find a complete listing of most of the keyboard related subroutines from my "Great Memory Map." Most of the routines not discussed in the article are primarily routines and memory addresses used by the major subroutines. If you are trying to understand how things work in the Model 100 ROM through a disassembly of the ROM, these addresses should be quite helpful.

So far I have discussed two of the three most used devices in the Model 100: the display and the keyboard. Next month, I'll discuss the RS-232/Modem and its ROM routines. PCM

TWO YEARS OF PCM

An Index to the Articles, Reviews and Authors Appearing in PCM from July 1983 through June 1985

Compiled and Edited by Leslie A. Foster

SUMMARY

In the period July 1983 to June 1985, PCM has published 321 articles in the following categories:
GENERAL - 17
MS-DOS - 60
MS-DOS - PRODUCT REVIEWS - 26
PORTABLE - 144
PORTABLE - EDUCATION - 5
PORTABLE - GAMES - 13
PORTABLE - PRODUCT REVIEWS - 56

Of the above articles, 234 were signed by authors. In addition, there were 27 programs for the Model 100 in bar code (denoted by ★ in the index).

GENERAL

Covington, Robert D. "Subroutine city: Searching." *(1985, June)* 54 —Three subroutines to search for strings of text.
Graham, Randy. "Communications: mule to modem." *(1985, April)* 27
Graham, Randy. "Comparison shopping in the electronic malls." *(1984, June)* 25 —Discussion of The Source.
Graham, Randy. "Dow Jones means business." *(1984, August)* 20 —Comments on Dow-Jones News Service.
Graham, Randy. "From mind games to a researcher's dream." *(1984, September)* 8 —Comments about information utilities.
Graham, Randy. "Learning to use the major information services." *(1984, October)* 9 —Comments about BRS, Dialog and Orbit.
Graham, Randy. "Let's go ahunting." *(1984, November)* 16 —Sample of searches on databases.
Graham, Randy. "Let's go exploring." *(1984, May)* 10 —Discussion of Compuserve.
Graham, Randy. "Orientation for newcomers." *(1985, May)* 15 —Introduction to the portables and MS-DOS computers.
Graham, Randy. "A retrospective look and an interesting anecdote." *(1985, March)* 46 —Wrap up on columns from previous year.
Graham, Randy. "Soapbox time for telecommunicators." *(1984, July)* 38 —Comments about security.
Graham, Randy. "Thumbing through the electronic card catalogs." *(1985, January)* 13 —Discussion of online databases.
Graham, Randy. "Walking through BRS/After dark." *(1984, December)* 37 —Sample of on-line search.
Graham, Randy. "What's fashionable this season in electronic communications?" *(1985, June)* 29
Hawk, Jim. "Dream portable... or reality?" *(1984, July)* 20 —The ideal portable described.
Hawk, Jim. "Nu-mail." *(1984, June)* 12 —Discussion of the various electronic mail vendors.
White, Richard A. "An introduction to spreadsheets." *(1985, June)* 10

MS-DOS

Barden, William, Jr. "Tandy, Epson, IBM and the great printer fiasco." *(1985, June)* 45 —How MS-DOS created problems for Tandy's printers.
Barden, William, Jr. "Windows and viewports on the 1000 and 1200." *(1985, May)* 56
Covington, Robert D. "File I/O techniques for your 1000, 1200 and 2000." *(1985, March)* 55
Covington, Robert D. "Implementing keyboard routines of power." *(1985, May)* 25 —Some useful subroutines.
Dischert, Dave ; and Keen, Dan. "Vector animation on the Tandy 1000." *(1985, April)* 46
Falk, Lawrence C. "Adorn your 2000's text screen with color." *(1985, February)* 33
Falk, Lawrence C. "Creating categorical directories." *(1984, July)* 19 —Organize files by subject on the 2000.
Falk, Lawrence C. "Personalizing your 2000's prompt." *(1984, November)* 35
Falk, Lawrence C. "Your 2000 can help access those applications." *(1984, August)* 24 —Tutorial on file manipulation.
Fetzko, Mark. "For pixel-packing artists, it's Quick-Draw." *(1984, November)* 21 —Graphics on the 2000.
Fisher, Seymour. "Let your computer locate those buried files." *(1985, May)* 55 —File catalog utility.
Humphress, Danny. "Adding, changing and rearranging with filters." *(1985, January)* 53 —MS-DOS tutorial
Humphress, Danny. "The adventures of BATchman!" *(1985, March)* 50 —Batch processing tips.
Humphress, Danny. "BATch files: the final chapter." *(1985, May)* 50
Humphress, Danny. "BATch man takes on formidable enemies." *(1985, April)* 30
Humphress, Danny. "A batch of commands." *(1985, February)* 13 —Execute complex commands with few keystrokes.
Humphress, Danny. "dBASE tutor part 1." *(1984, June)* 15 —Series on dBASE II.
Humphress, Danny. "dBASE tutor part 2 — more power to you." *(1984, July)* 24
Humphress, Danny. "dBASE tutor part 3 — files of a different sort." *(1984, August)* 32
Humphress, Danny. "dBASE tutor part 4 - creating reports with the 'sophisticated system.'" *(1984, October)* 15
Humphress, Danny. "dBASE tutor part 5 — database maintenance." *(1984, November)* 25
Humphress, Danny. "dBASE tutor part 6 — building a foundation for application programs." *(1984, December)* 23
Humphress, Danny. "dBASE tutor part 7 - files for two." *(1985, January)* 47
Humphress, Danny. "dBASE tutor part 8 — the operational calculating file drawer." *(1985, February)* 19
Humphress, Danny. "dBASE tutor part 9 — of strings and things." *(1985, March)* 53
Humphress, Danny. "dBASE tutor part 10 — all systems functioning." *(1985, April)* 21
Humphress, Danny. "dBASE tutor part 11 — the dProgramming zone." *(1985, May)* 44
Humphress, Danny. "dBASE tutor part 12 — screen formats: better than plain vanilla." *(1985, June)* 21
Humphress, Danny. "A few good bars (cover charge only)." *(1984, August)* 22 —Bar graph drawing.
Humphress, Danny. "The hierarchal tree." *(1985, June)* 17 —Tutorial on MS-DOS's hierarchal file directories.
Humphress, Danny. "Mastering MS-DOS part 1." *(1984, May)* 8 —As applied on the Model 2000.
Humphress, Danny. "Mastering MS-DOS part 2." *(1984, June)* 8
Humphress, Danny. "Mastering MS-DOS part 3 — exploring MS-DOS commands." *(1984, July)* 27
Humphress, Danny. "Mastering MS-DOS part 4." *(1984, August)* 16
Humphress, Danny. "Mastering MS-DOS part 5 — Edlin function keys." *(1984, September)* 33
Humphress, Danny. "Mastering MS-DOS part 6 — more on Edlin." *(1984, October)* 24
Humphress, Danny. "Mastering MS-DOS part 7 — a final look at Edlin." *(1984, November)* 14
Humphress, Danny. "The perfect companion." *(1984, April)* 32 —Review of the Model 2000.
Humphress, Danny. "Plumbing for beginners." *(1984, December)* 30 —MS-DOS tutorial.
Humphress, Danny. "The Tandy 1000: Tandy's surprise machine." *(1985, February)* 53
Humphress, Danny. "Two triple threats for the 2000: a comparative review of Lotus 1-2-3 and SuperCalc." *(1984, September)* 42
Pifer, David R. "Making pies a piece of cake." *(1984, September)* 11 —Pie charts on the 2000.
Pifer, David R. "Skedit - part 1." *(1984, December)* 8 —A full featured graphics editor for the Model 2000.
Pifer, David R. "Skedit - part 2." *(1985, January)* 15
Pifer, David R. "Skedit - part 3." *(1985, February)* 44
Pifer, David R. "Skedit - part 4." *(1985, March)* 34
Rideout, Ralph. "Translate plotting tedium into ease with Graphplot 2000." *(1984, October)* 12
Sanders, Wayne. "Flag of the month club - Australia." *(1984, November)* 36
Sanders, Wayne. "Flag of the month club - Canada." *(1984, June)* 11
Sanders, Wayne. "Flag of the month club - Great Britain." *(1984, August)* 36
Sanders, Wayne. "Flag of the month club - South Korea." *(1984, September)* 36
Sanders, Wayne. "Flag of the month club- Olympic games." *(1984, July)* 37
Sanders, Wayne. "The gallery." *(1985, January)* 35 —Graphics demo.
Sanders, Wayne. "The gallery." *(1985, February)* 52
Sanders, Wayne. "The gallery." *(1985, March)* 12
Sanders, Wayne. "The gallery." *(1985, April)* 42
Sanders, Wayne. "The gallery." *(1985, May)* 31
Sanders, Wayne. "The gallery." *(1985, June)* 42
Sanders, Wayne. "Those great (and colorful) '2000' graphics!" *(1984, May)* 37 —Draw a U.S. flag.
Sanders, Wayne. "2000 gallery." *(1984, December)* 16 —Graphics demo.

MS-DOS - PRODUCT REVIEWS

"Accounting partner." *(1985, February)* 59
"Accounting partner." *(1985, March)* 71
"Condor3." *(1985, February)* 56
"filePro." *(1984, September)* 44
"filePro." *(1985, June)* 69
"Freeform (database)." *(1984, December)* 49
"Home accountant plus." *(1985, April)* 62
"LaserJet printer." *(1985, June)* 63
"MAI Accounts Receivable." *(1984, August)* 47
"MAI General Ledger." *(1985, May)* 68
"Mailer." *(1985, June)* 65
"MultiMate." *(1984, November)* 43
"Open access." *(1984, August)* 43
"PertMaster." *(1985, January)* 58
"pfs:write." *(1985, March)* 66
"Power manager." *(1985, March)* 68
"Real estate investor II." *(1984, November)* 41
"Small business management system." *(1985, June)* 70
"Snake." *(1985, May)* 67
"Telecommuter." *(1985, January)* 57
"Telecommuter." *(1985, March)* 73
"Teleterm." *(1984, October)* 44
"Turbo Pascal." *(1985, May)* 66
"The Tutor (book)." *(1985, April)* 60
"Volkswriter deluxe." *(1984, December)* 46
"The 1985 computer buying guide." *(1985, April)* 66

PORTABLE

Balonis, Ronald F. "Know the size of your files and programs at a glance." *(1985, May)* 33 —Bar code listing.★
Balonis, Ronald F. "Test your intuitive decision making skills." *(1985, April)* 53 —Decision making program. Bar code listing.★
Boulet, Larry. "A timer for your chess game." *(1983, September)* 15
Cassidy, Gene ; and Blue, Bill. "Tandy apples: the Model 100 Apple connection." *(1984, January)* 24 —Cable to connect to Apple II+.
Cornman, Aileen ; and Cornman, John. "Home inventory: Model 100 portability makes the job easier." *(1984, July)* 12
Cornman, Aileen ; and Cornman, John. "The PoCo word processor." *(1985, February)* 37 —Word processing program. Bar code listing.★
Cornman, Aileen ; and Cornman, John. "What's not in the Model 100 manual?" *(1984, October)* 21 —Fixes up errors in the manual.
Covington, Robert D. "The portable machine - part 1." *(1984, December)* 34 —Machine language tutorial.
Covington, Robert D. "The portable machine - part 2." *(1985, January)* 38
Covington, Robert D. "The portable machine - part 3." *(1985, March)* 25
Covington, Robert D. "The portable machine - part 4." *(1985, April)* 31
Covington, Robert D. "The portable machine - part 5." *(1985, June)* 31 —Tutorial about the Model 100 display memory map.
Craven, Richard. "Logging account time with 'Time Clock.'" *(1984, September)* 25 —Bar code listing.★ Correction, January 1985, page 62.
Creed, David W. "The freedom machine." *(1984, April)* 25 —Use of Model 100 in the newsroom.
Delbourgo, Bob. "Of prime importance." *(1984, May)* 27 —Prime numbers tutorial.

Delbourgo, Tino. "You'll DO better with this TEXT utility." *(1984, February)* 34 —Aid in listing Basic programs.

DiStefano, Tony. "Inside the Model 100 - part 1." *(1984, December)* 44 —Discussion of the workings of the 100.

DiStefano, Tony. "Inside the Model 100 - part 2." *(1985, January)* 31

DiStefano, Tony. "Inside the Model 100 - part 3." *(1985, March)* 21

DiStefano, Tony. "Inside the Model 100 - part 4." *(1985, April)* 56

DiStefano, Tony. "Inside the Model 100 - part 5." *(1985, May)* 10

DiStefano, Tony. "Installing rechargeable batterles in your Model 100." *(1984, September)* 31

Donohue, Joseph. "Keeping an eye on home heating cost." *(1983, December)* 24

Downard, Dan. "Biorhythyms: plotting your cycles." *(1983, August)* 8

Ellers, Ed. "The Disk/Video interface." *(1984, March)* 16

Falk, Lawrence C. "Let's get organized with our Model 100." *(1983, August)* 20 —Discussion of ADDRSS and SCHEDL.

Falk, Lawrence C. "The Tandy 200." *(1985, January)* 8

"File sorting can be good, but beware of pitfalls." *(1983, July)* 20

Flamer, Thomas J. "Analyze your data with linear curve fitting." *(1983, September)* 12

Frowenfeld, Robert. "A bit of latitude." *(1983, November)* 10 —Aid in surveying.

Frowenfeld, Robert. "Build your own null modem adapter." *(1984, December)* 18

Frowenfeld, Robert. "CHECK.BA—don't leave home without it!" *(1984, October)* 27 —Checkbook balancer.

Frowenfeld, Robert. "Computer to computer communications with TELCOM." *(1983, October)* 24

Frowenfeld, Robert. "Dash away all!" *(1983, December)* 22 —Generate the Morse code.

Frowenfeld, Robert. "Expense reporting in search of the 'round toit.'" *(1984, May)* 15 —Expense monitor program. Bar code listing.★

Frowenfeld, Robert. "A graph is worth a thousand digits." *(1984, July)* 8 —How to draw bar graphs. Bar code listing.★

Frowenfeld, Robert. "Logging your paycheck can be helpful down the road." *(1984, August)* 37 —Home finance.

Frowenfeld, Robert. "The music box." *(1983, September)* 8 —Music on the Model 100."

Frowenfeld, Robert. "A new stock option: monitoring daily fluctuations with TRACK.BA." *(1984, April)* 12

Frowenfeld, Robert. "On the road." *(1983, July)* 27 —Applications - amortization program.

Frowenfeld, Robert. "On the road." *(1983, August)* 26 —Keeps track of gas mileage, etc. of car.

Frowenfeld, Robert. "On the road." *(1983, September)* 24 —Amortization program.

Frowenfeld, Robert. "Out of checks? Print your own!" *(1983, October)* 12 —Aid in writing checks.

Frowenfeld, Robert. "Planning your car's maintenance program (or doing what you AUTO.DO)." *(1984, February)* 12

Frowenfeld, Robert. "Plotting graphs with the Model 100." *(1983, July)* 8

Frowenfeld, Robert. "Road weary? Dial a KISS!" *(1984, November)* 38 —Radio station names and frequencies. Correction, January 1985, page 62.

Frowenfeld, Robert. "Stocking up on reinvested dividends." *(1984, September)* 38 —Keep track of your stock dividends.

Frowenfeld, Robert. "A summer exchange program." *(1984, June)* 32 —Foreign currency calculator.

Frowenfeld, Robert. "Taxes, Do? Taxes, Bah!" *(1984, January)* 8 —Aid in tax record keeping.

Frowenfeld, Robert. "A time to take stock." *(1983, December)* 12 —Stock portfolio manager.

Frowenfeld, Robert. "Travelling to the aisles." *(1983, October)* 9 —Assist in grocery shopping.

"Glancing at the new 100's version of Basic." *(1983, July)* 18

"Government computer expo: showtime in federal city." *(1984, May)* 19

Graham, Randy. "Greetings from a small computer." *(1984, April)* 28 —Tutorial on TELCOM.

Graham, Randy. "Reaching out with TELCOM." *(1984, March)* 24 —Hints on how to use TELCOM effectively.

Guscott, J. Kenneth. "The executive decoder ring: a different approach to file security." *(1984, August)* 11 —Put your files in code. Bar code listing.★

Guscott, J. Kenneth. "Five easy pieces for PoCo." *(1984, May)* 20 —A music program. Bar code listing.★

Hanson, Donald J. "Prepare your next expense account with EXPNSE." *(1983, November)* 25

Hawk, Jim. "Are portables safe for flying?" *(1983, November)* 16 —Controversy about portables in airplanes.

Hawk, Jim. "Do aircraft interfere with portables?" *(1984, October)* 19

Hawk, Jim. "Getting the most from your Model 100/200." *(1985, May)* 21 —Hints for newcomers.

Hawk, Jim. "A magic wand for your Model 100." *(1983, December)* 28 —Bar code tutorial.

Hawk, Jim. "Mass storage: the software and hardware arrive." *(1984, May)* 39 —Report on Disk+ and Holmes Wafer drive.

Hawk, Jim. "The Model 100 goes to Washington: A journalistic revolution." *(1984, November)* 8

Hawk, Jim. "News from SCPIADCA." *(1984, January)* 30 —Update on use of portables in airplanes.

Hawk, Jim. "PoCo passes preliminary tests with flying colors." *(1984, April)* 22 —More on portables in airplanes.

Hawk, Jim. "A PoCo performance that is out of this world." *(1985, March)* 14 —Model 100 on the space shuttle.

Hawk, Jim. "Portable mass storage is here." *(1984, February)* 16 —Discussion on disks.

Hawk, Jim. "Travels with PoCo: personal discoveries along the way." *(1984, March)* 19 —Hints for newcomers.

Hawk, Jim. "Try this byte-counting utility—just for size." *(1984, April)* 14 —Size of each file.

Hawk, Jim. "200 - a new generation portable is born." *(1985, April)* 25

Hawk, Jim. "The 96K Model 100—or how to fit two Model 100's in the same box." *(1984, September)* 18

Hogg, Frank. "Found! The M100's hidden cursor control keys." *(1983, September)* 22

Humphress, Danny. "Bar code Basic." *(1984, April)* 17 —Bar code reading program. Bar code listing.★

Humphress, Danny. "Beam me up Scotty—in ASCII of course." *(1984, May)* 13 —Transferring files from Model 100 to Model 2000.

Humphress, Danny. "The revised reader." *(1984, May)* 25 —Updated bar code program. Bar code listing.★

Humphress, Danny. "Scanning the pages of PCM." *(1985, January)* 27 —Updated bar code program. Bar code listing.★

Ireland, Nathaniel. "The portable vacation." *(1985, June)* 26 —Keep running records of vacation expenses. Bar code listing.★

Ireland, Nathaniel F. "A Model 100 calculator." *(1984, June)* 35 —Use it as a calculator. Bar code listing.★

Ireland, Nathaniel F. "Navloc. *(1984, June)* 30 —Determine latitude and longtitude.

Ireland, Nathaniel F. "Short-term memory booster!" *(1984, March)* 32 —Appointment calendar.

Kintsfather, David. "The portable computer as a social science research tool." *(1984, September)* 17 —Comments about statistics.

Klages, Jon P. "3-D for your LCD." *(1984, August)* 30 —3-D graphics. Bar code listing.★

Kleinfeld, Andrew J. "Dump it on Compuserve (and set yourself free)." *(1984, January)* 14 —Use Compuserve to hold your files.

"Little computer demo." *(1983, October)* 22 —Graphics demo.

Lord, Vincent, Jr. "Determine your return on investments." *(1985, March)* 10 —Bar code listing.★

Lord, Vincent, Jr. "Determining gross profit margins." *(1983, August)* 7

Lord, Vincent, Jr. "Financial ratios can help determine corporate 'soundness.'" *(1983, July)* 14

Lord, Vincent, Jr. "Quick menu—a programming utility." *(1984, April)* 27 —Add menus to Basic programs. Bar code listing.★

Lord, Vincent, Jr. "Sailing the Mediterranean with PoCo." *(1983, October)* 16 —Use Model 100 for navigating.

Lord, Vincent, Jr. "Three methods of property depreciation." *(1984, January)* 28

Mendenhall, R. E. "Decisions, decisions—using your Model 100 as a decision making aid." *(1984, October)* 37

Mendenhall, R. E. "It's a date!" *(1985, May)* 53 —Calendar calculations. Bar code listing.★

Mitchell, D. E. "The PoCo-CoCo connection." *(1984, August)* 26 —Transfer files to the Color Computer.

Nolan, Bill. "Examining Basic's INSTR function." *(1984, May)* 35 —Basic tutorial.

Nolan, Bill. "PoCodometer." *(1984, October)* 41 —Speedometer check. Bar code listing.★

Oppedahl, Carl. "Give your Model 100 an autoanswer modem for under $10." *(1984, November)* 30

Oppedahl, Carl. "Going online at the office with PoCo." *(1984, July)* 22 —Direct connect to multiline office phone.

Oppedahl, Carl. "How to send graphics characters to a dot matrix printer." *(1985, May)* 39 —Bar code listing.★

Paludan, Ronald. "Appending PoCo files." *(1983, November)* 24

Paludan, Ronald. "How to make your RAM files invisible." *(1984, February)* 29 —Files will not appear on the menu.

Paludan, Ronald. "The PoCo calendar." *(1983, December)* 18 —Print calendars on the screen.

Petersen, Cheryl. "Whose skies are user friendly?" *(1984, April)* 22 —More on portables in airplanes.

Preble, Laurence D. "Get 'hands on' experience with this keyboard sender." *(1984, January)* 16 —Use Model 100 to send Morse Code.

Preble, Laurence D. "Seller-friendly." *(1984, December)* 40 —Order form and invoices. Bar code listing.★

"RAMbulance service." *(1984, January)* 33 —Application of Model 100 in an ambulance.

Randall, Larry. "Automatic line numbering for Basic." *(1984, March)* 30

Randall, Larry. "Let your portable list those variables." *(1984, February)* 30 —Basic programming utility.

Randall, Larry. "Looking for ROM." *(1984, January)* 19 —Disassembler to look at the memory.

Randall, Larry. "A renumbering routine for PoCo." *(1983, December)* 32 —Renumber Basic programs.

"Random function will require some figuring." *(1983, August)* 13 —Discussion of the RND function.

Rogers, Richard T. "Snoop-proof your memory!" *(1983, December)* 33 —Add a password protection.

Rollins, Willis. "Examining your machine's capabilities." *(1984, February)* 32 —Tutorial on the Model 100.

Rosen, Bob. "Inside and outside." *(1983, August)* 22 —General overview of Model 100.

Rosen, Bob. "Telecommunications and the Model 100." *(1983, July)* 13

Rothstein, Rick. "Screen dump 100." *(1985, January)* 43 —Graphics screen dump program. Bar code listing.★

Schwartz, Steven A. "The exchange factor." *(1985, April)* 49 —Foreign currency converter. Bar code listing.★

Sissala, Gary A. "Create letter perfect text files with this Print Format Program." *(1983, October)* 34 —Improves the TEXT output.

Socci, Vance. "DIALM8 for a handy autodialing extender." *(1983, November)* 19 —Improvements to ADRS program.

Stanford, Michael. "Drawing a pie chart on the screen." *(1983, August)* 24

Stauffer, William T. "Disk video interface: a user's critique." *(1984, May)* 17 —Includes how to install second drive.

Stauffer, William T. "Homemade PoCo to CCR cable." *(1983, October)* 32

Stauffer, William T. "Six financial programs for PoCo." *(1984, February)* 24 —Amortization, etc.

Stevens, Guerri F. "Download aid." *(1983, October)* 26

Trevor, Alexander. "Weather download—the lowdown." *(1983, December)* 16 —Download the Vidtex radar maps from Compuserve.

Wheeler, Dale. "Among printer utilities LLST is without parallel." *(1983, September)* 21 —Utility to use serial interface to printer.

Wheeler, Dale. "Organize your ADRS.DO files with this 'label maker.'" *(1983, November)* 14 — (Correction, December, 1983, page 37)

Wheeler, Dale. "Pruning your programs." *(1984, January)* 26 —Basic utility

White, Richard A. "A Basic start." *(1983, October)* 6 —Tutorial on Basic.

White, Richard A. "The bottom line: settling up the expense account." *(1985, April)* 10 —Wrap-up of expense account program.

White, Richard A. "Dealing with Basic string functions." *(1984, August)* 8 —Tutorial on Basic.

White, Richard A. "Gaining sophistication in printing techniques." *(1984, March)* 8 —Tutorial on Basic.

White, Richard A. "Getting a line on your computer's operators." *(1984, September)* 27 —Tutorial on Basic.

White, Richard A. "Groundwork for future file handling." *(1984, April)* 9 —Tutorial on Basic.

White, Richard A. "Housekeeping for the expense account." *(1985, March)* 61

White, Richard A. "Keeping the expense account." *(1985, January)* 50 —Tutorial with practical example.

White, Richard A. "More on variables." *(1984, July)* 34 —Tutorial on Basic.

White, Richard A. "Program control." *(1983, December)* 8 —Tutorial on Basic.

White, Richard A. "Program logic: a discussion in context." *(1984, October)* 33 —Tutorial on Basic. Bar code listing.★

White, Richard A. "Programming to save memory." *(1984, January)* 12 —Tutorial on Basic.

White, Richard A. "Setting up the expense account." *(1985, February)* 26

White, Richard A. "Some 'constants' about variables." *(1983, November)* 8 —Tutorial on Basic.

White, Richard A. "Techniques for handling data." *(1984, February)* 8 —Tutorial on Basic.

Zucconi, George. "Data and file transfer between the Model 100 and CP/M computers." *(1985, February)* 9

PORTABLE - EDUCATION

Delbourgo, Bob. "A summing up—symbolically." *(1984, September)* 15 —Math exercises. Major correction, January 1985, page 62.

Frowenfeld, Robert. "A capital quiz for the wordly computerist." *(1984, March)* 13 —Capitals and countries

Perkins, Mel. "Meet Professor PoCo." *(1984, July)* 32 —Math drill. Bar code listing.★

Perkins, Mel. "Spelling practice." *(1984, February)* 11

"This test will help your kids." *(1983, July)* 24 —Drill for children.

PORTABLE - GAMES

Albrecht, Bob ; and Albrecht, Karl. "Executive recreations: twinkle, twinkle, guess what?" *(1985, March)* 43

Albrecht, Bob ; and Albrecht, Karl. "Executive recreations—getting your Wordsworth." *(1984, December)* 28 —Series on "computer related diversions."

Albrecht, Bob ; and Albrecht, Karl. "Executive recreations—how much are these words worth?" *(1985, January)* 55

Delbourgo, Bob. "Keyboard Dominoes." *(1984, November)* 11 —Bar code listing.★

Delbourgo, Bob. "Solo—solitaire with spice." *(1984, February)* 22

Firschein, Ben. "Zoid patrol." *(1984, June)* 37

Frowenfeld, Robert. "Try your hand at Blackjack—it's a good deal for dull moments!" *(1983, August)* 18

Lord, Vincent, Jr. "Let's go spelunking but watch out for the Wumpus." *(1983, October)* 29

"Match wits with Nim." *(1983, July)* 22

Paludan, Ronald. "Try this triangle challenge." *(1983, September)* 18 —The game of SIM.

Roberson, Larry M. "You can 'playit' on your PoCo." *(1984, September)* 21 —Bar code listing.★

Rothstein, Rick. "PoCo and the bandit." *(1985, February)* 22 —Slot machine game.

Rothstein, Rick. "Square deal." *(1984, June)* 21 — Version of poker solitaire. Bar code listing.★

PORTABLE - PRODUCT REVIEWS

"Assembler.Co." *(1985, May)* 70

"Astro." *(1985, May)* 71

"Autopen." *(1984, July)* 42

"BOOK (bookkeeping)." *(1984, March)* 37

"Bar code reader (Radio Shack)." *(1984, March)* 35

"Bi-tech bar code reader." *(1983, December)* 35

"Black Jack (modem interface)." *(1985, April)* 61

"Blockade." *(1983, July)* 29

"Business finance." *(1985, February)* 58

"Businesspak+." *(1983, August)* 29

"Businesspak+." *(1983, September)* 29 —More of the programs discussed here.

"CCR-82 (cassette recorder)." *(1985, January)* 61

"Code keeper 3.0." *(1985, April)* 64

"Cryptease." *(1985, June)* 69

"DMP-20 printer." *(1984, January)* 35

"Data+ and Sort2+." *(1983, December)* 35

"Disk+ ROM." *(1984, October)* 43

"Disk-buddy." *(1985, March)* 70

"Financial analysis package 1." *(1984, February)* 37

"Frankenstein adventure." *(1983, September)* 30

"FullVu." *(1984, September)* 46

"Greymoon." *(1983, November)* 37

"How to do it on the TRS-80 (book)." *(1984, January)* 34

"The Ideabook." *(1984, August)* 46

"Lucid (spreadsheet)." *(1984, December)* 48

"Melody maker." *(1985, May)* 71

"Model100 Assembler." *(1984, May)* 43

"NavComp 1.0." *(1984, May)* 42

"Paper tractor." *(1984, February)* 36

"Porta calc." *(1983, November)* 36

"PortaPac 100." *(1984, December)* 50

"PortaStat." *(1984, January)* 36

"PortaTax." *(1984, March)* 36

"Portamax+." *(1984, January)* 35

"Purple Computing memory expansion." *(1984, April)* 37

"Quick plot." *(1983, December)* 36

"ROM 2." *(1985, June)* 68

"The Radio Shack Notebook Computer (book)." *(1984, October)* 43

"Renumber." *(1984, November)* 45

"Scribe 2.0." *(1984, March)* 37

"Scribe." *(1983, October)* 37

"Scripsit 100." *(1984, August)* 46

"Sky raider." *(1984, June)* 43

"Spectaculator." *(1984, October)* 46

"Spectrum upgrade." *(1983, November)* 37

"TPLAN/NPLAN spreadsheet." *(1984, July)* 40

"T-BASE." *(1984, June)* 42

"Text power 100." *(1985, February)* 56

"ThinPrint 80." *(1984, August)* 48

"ThinStore." *(1984, December)* 51

"ThinType." *(1985, January)* 58

"Travelling sales manager." *(1984, July)* 43

"Travelling appointment manager." *(1984, August)* 44

"User guide and applications for the Model 100 (book)." *(1984, November)* 43

"Write+." *(1984, April)* 36

"60 business applications for the TRS-80 Model 100 computer." *(1985, April)* 66

AUTHOR INDEX

Albrecht, Bob ; and Albrecht, Karl. "Executive recreations: twinkle, twinkle, guess what?" *(1985, March)* 43

Albrecht, Bob ; and Albrecht, Karl. "Executive recreations—getting your Wordsworth." *(1984, December)* 28 —Series on "computer related diversions."

Albrecht, Bob ; and Albrecht, Karl. "Executive recreations—how much are these words worth?" *(1985, January)* 55

Balonis, Ronald F. "Know the size of your files and programs at a glance." *(1985, May)* 33 —Bar code listing.★

Balonis, Ronald F. "Test your intuitive decision making skills." *(1985, April)* 53 —Decision making program. Bar code listing.★

Barden, William, Jr. "Tandy, Epson, IBM and the great printer fiasco." *(1985, June)* 45 —How MS-DOS created problems for Tandy's printers.

Barden, William, Jr. "Windows and viewports on the 1000 and 1200." *(1985, May)* 56

Boulet, Larry. "A timer for your chess game." *(1983, September)* 15

Cassidy, Gene ; and Blue, Bill. "Tandy apples: the Model 100 Apple connection." *(1984, January)* 24 —Cable to connect to Apple II+.

Cornman, Aileen ; and Cornman, John. "Home inventory: Model 100 portability makes the job easier." *(1984, July)* 12

Cornman, Aileen ; and Cornman, John. "The PoCo word processor." *(1985, February)* 37 —Word processing program. Bar code listing.★

Cornman, Aileen ; and Cornman, John. "What's not in the Model 100 manual?" *(1984, October)* 21 —Fixes up errors in the manual.

Covington, Robert D. "File I/O techniques for your 1000, 1200 and 2000." *(1985, March)* 55

Covington, Robert D. "Implementing keyboard routines of power." *(1985, May)* 25 —Some useful subroutines.

Covington, Robert D. "The portable machine - part 1." *(1984, December)* 34 —Machine language tutorial.

Covington, Robert D. "The portable machine - part 2." *(1985, January)* 38

Covington, Robert D. "The portable machine - part 3." *(1985, March)* 25

Covington, Robert D. "The portable machine - part 4." *(1985, April)* 31

Covington, Robert D. "The portable machine - part 5." *(1985, June)* 31 —Tutorial about the Model 100 display memory map.

Covington, Robert D. "Subroutine city: Searching." *(1985, June)* 54 —Three subroutines to search for strings of text.

Craven, Richard. "Logging account time with 'Time Clock.'" *(1984, September)* 25 —Bar code listing.★ Correction, January 1985, page 62.

Creed, David W. "The freedom machine." *(1984, April)* 25 —Use of Model 100 in the newsroom.

Delbourgo, Bob. "Keyboard Dominoes." *(1984, November)* 11 —Bar code listing.★

Delbourgo, Bob. "Of prime importance." *(1984, May)* 27 —Prime numbers tutorial.

Delbourgo, Bob. "Solo—solitaire with spice." *(1984, February)* 22

Delbourgo, Bob. "A summing up—symbolically." *(1984, September)* 15 —Math exercises. Major correction, January 1985, page 62.

Delbourgo, Tino. "You'll DO better with this TEXT utility." *(1984, February)* 34 —Aid in listing Basic programs.

DiStefano, Tony. "Inside the Model 100 - part 1." *(1984, December)* 44 —Discussion of the workings of the 100.

DiStefano, Tony. "Inside the Model 100 - part 2." *(1985, January)* 31

DiStefano, Tony. "Inside the Model 100 - part 3." *(1985, March)* 21

DiStefano, Tony. "Inside the Model 100 - part 4." *(1985, April)* 56

DiStefano, Tony. "Inside the Model 100 - part 5." *(1985, May)* 10

DiStefano, Tony. "Installing rechargeable batteries in your Model 100." *(1984, September)* 31

Dischert, Dave ; and Keen, Dan. "Vector animation on the Tandy 1000." *(1985, April)* 46

Donohue, Joseph. "Keeping an eye on home heating cost." *(1983, December)* 24

Downard, Dan. "Biorhythyms: plotting your cycles." *(1983, August)* 8

Ellers, Ed. "The Disk/Video interface." *(1984, March)* 16

Falk, Lawrence C. "Adorn your 2000's text screen with color." *(1985, February)* 33

Falk, Lawrence C. "Creating categorical directories." *(1984, July)* 19 —Organize files by subject on the 2000.

Falk, Lawrence C. "Let's get organized with our Model 100." *(1983, August)* 20 —Discussion of ADDRSS and SCHEDL.

Falk, Lawrence C. "Personalizing your 2000's prompt." *(1984, November)* 35

Falk, Lawrence C. "The Tandy 200." *(1985, January)* 8

Falk, Lawrence C. "Your 2000 can help access those applications." *(1984, August)* 24 —Tutorial on file manipulation.

Fetzko, Mark. "For pixel-packing artists, it's Quick-Draw." *(1984, November)* 21 —Graphics on the 2000.

Firschein, Ben. "Zoid patrol." *(1984, June)* 37

Fisher, Seymour. "Let your computer locate those buried files." *(1985, May)* 55 —File catalog utility.

Flamer, Thomas J. "Analyze your data with linear curve fitting." *(1983, September)* 12

Frowenfeld, Robert. "A bit of latitude." *(1983, November)* 10 —Aid in surveying.

Frowenfeld, Robert. "Build your own null modem adapter." *(1984, December)* 18

Frowenfeld, Robert. "CHECK.BA—don't leave home without it!" *(1984, October)* 27 —Checkbook balancer.

Frowenfeld, Robert. "A capital quiz for the wordly computerist." *(1984, March)* 13 —Capitals and countries

Frowenfeld, Robert. "Computer to computer communications with TELCOM." *(1983, October)* 24

Frowenfeld, Robert. "Dash away all!" *(1983, December)* 22 —Generate the Morse code.

Frowenfeld, Robert. "Expense reporting in search of the 'round toit.'" *(1984, May)* 15 —Expense monitor program. Bar code listing.★

Frowenfeld, Robert. "A graph is worth a thousand digits." *(1984, July)* 8 —How to draw bar graphs. Bar code listing.★

Frowenfeld, Robert. "Logging your paycheck can be helpful down the road." *(1984, August)* 37 —Home finance.

Frowenfeld, Robert. "The music box." *(1983, September)* 8 —Music on the Model 100."

Frowenfeld, Robert. "A new stock option: monitoring daily fluctuations with TRACK.BA." *(1984, April)* 12

Frowenfeld, Robert. "On the road." *(1983, July)* 27 —Applications - amortization program.

Frowenfeld, Robert. "On the road." *(1983, August)* 26 —Keeps track of gas mileage, etc. of car.

Frowenfeld, Robert. "On the road." *(1983, September)* 24 —Amortization program.

Frowenfeld, Robert. "Out of checks? Print your own!" *(1983, October)* 12 —Aid in writing checks.

Frowenfeld, Robert. "Planning your car's maintenance program (or doing what you AUTO.DO)." *(1984, February)* 12

Frowenfeld, Robert. "Plotting graphs with the Model 100." *(1983, July)* 8

Frowenfeld, Robert. "Road weary? Dial a KISS!" *(1984, November)* 38 —Radio station names and frequencies. Correction, January 1985, page 62.

Frowenfeld, Robert. "Stocking up on reinvested dividends." *(1984, September)* 38 —Keep track of your stock dividends.

Frowenfeld, Robert. "A summer exchange program." *(1984, June)* 32 —Foreign currency calculator.

Frowenfeld, Robert. "Taxes, Do? Taxes, Bah!" *(1984, January)* 8 —Aid in tax record keeping.

Frowenfeld, Robert. "A time to take stock." *(1983, December)* 12 —Stock portfolio manager.

Frowenfeld, Robert. "Travelling to the aisles." *(1983, October)* 9 —Assist in grocery shopping.

Frowenfeld, Robert. "Try your hand at Blackjack—it's a good deal for dull moments!" *(1983, August)* 18

Graham, Randy. "Communications: mule to modem." *(1985, April)* 27

Graham, Randy. "Comparison shopping in the electronic malls." *(1984, June)* 25 —Discussion of The Source.

Graham, Randy. "Dow Jones means business." *(1984, August)* 20 —Comments on Dow-Jones News Service.

Graham, Randy. "From mind games to a researcher's dream." *(1984, September)* 8 —Comments about information utilities.

Graham, Randy. "Greetings from a small computer." *(1984, April)* 28 —Tutorial on TELCOM.

Graham, Randy. "Learning to use the major information services." *(1984, October)* 9 —Comments about BRS, Dialog and Orbit.

Graham, Randy. "Let's go ahunting." *(1984, November)* 16 —Sample of searches on databases.

Graham, Randy. "Let's go exploring." *(1984, May)* 10 —Discussion of Compuserve.

Graham, Randy. "Orientation for newcomers." *(1985, May)* 15 —Introduction to the portables and MS-DOS computers.

Graham, Randy. "Reaching out with TELCOM." *(1984, March)* 24 —Hints on how to use TELCOM effectively.

Graham, Randy. "A retrospective look and an interesting anecdote." *(1985, March)* 46 —Wrap up on columns from previous year.

Graham, Randy. "Soapbox time for telecommunicators." *(1984, July)* 38 —Comments about security.

Graham, Randy. "Thumbing through the electronic card catalogs." *(1985, January)* 13 —Discussion of online databases.

Graham, Randy. "Walking through BRS/After dark." *(1984, December)* 37 —Sample of on-line search.

Graham, Randy. "What's fashionable this season in electronic communications?" *(1985, June)* 29

Guscott, J. Kenneth. "The executive decoder ring: a different approach to file security." *(1984, August)* 11 —Put your files in code. Bar code listing.★

Guscott, J. Kenneth. "Five easy pieces for PoCo." *(1984, May)* 20 —A music program. Bar code listing.★

Hanson, Donald J. "Prepare your next expense account with EXPNSE." *(1983, November)* 25

Hawk, Jim. "Are portables safe for flying?" *(1983, November)* 16 —Controversy about portables in airplanes.

Hawk, Jim. "Do aircraft interfere with portables?" *(1984, October)* 19

Hawk, Jim. "Dream portable... or reality?" *(1984, July)* 20 —The ideal portable described.

Hawk, Jim. "Getting the most from your Model 100/200." *(1985, May)* 21 —Hints for newcomers.

Hawk, Jim. "A magic wand for your Model 100." *(1983, December)* 28 —Bar code tutorial.

Hawk, Jim. "Mass storage: the software and hardware arrive." *(1984, May)* 39 —Report on Disk+ and Holmes Wafer drive.

Hawk, Jim. "The Model 100 goes to Washington: A journalistic revolution." *(1984, November)* 8

Hawk, Jim. "News from SCPIADCA." *(1984, January)* 30 —Update on use of portables in airplanes.

Hawk, Jim. "Nu-mail." *(1984, June)* 12 —Discussion of the various electronic mail vendors.

Hawk, Jim. "PoCo passes preliminary tests with flying colors." *(1984, April)* 22 —More on portables in airplanes.

Hawk, Jim. "A PoCo performance that is out of this world." *(1985, March)* 14 —Model 100 on the space shuttle.

Hawk, Jim. "Portable mass storage is here." *(1984, February)* 16 —Discussion on disks.

Hawk, Jim. "Travels with PoCo: personal discoveries along the way." *(1984, March)* 19 —Hints for newcomers.

Hawk, Jim. "Try this byte-counting utility—just for size." *(1984, April)* 14 —Size of each file.

Hawk, Jim. "200 - a new generation portable is born." *(1985, April)* 25

Hawk, Jim. "The 96K Model 100—or how to fit two Model 100's in the same box." *(1984, September)* 18

Hogg, Frank. "Found! The M100's hidden cursor control keys." *(1983, September)* 22

Humphress, Danny. "Adding, changing and rearranging with filters." *(1985, January)* 53 —MS-DOS tutorial

Humphress, Danny. "The adventures of BATchman!" *(1985, March)* 50 —Batch processing tips.

Humphress, Danny. "BATch files: the final chapter." *(1985, May)* 50

Humphress, Danny. "BATch man takes on formidable enemies." *(1985, April)* 30

Humphress, Danny. "Bar code Basic." *(1984, April)* 17 —Bar code reading program. Bar code listing.★

Humphress, Danny. "A batch of commands." *(1985, February)* 13 —Execute complex commands with few keystrokes.

Humphress, Danny. "Beam me up Scotty—in ASCII of course." *(1984, May)* 13 —Transferring files from Model 100 to Model 2000.

Humphress, Danny. "dBASE tutor part 1." *(1984, June)* 15 —Series on dBASE II.

Humphress, Danny. "dBASE tutor part 2 — more power to you." *(1984, July)* 24

Humphress, Danny. "dBASE tutor part 3 — files of a different sort." *(1984, August)* 32

Humphress, Danny. "dBASE tutor part 4 - creating reports with the 'sophisticated system.'" *(1984, October)* 15

Humphress, Danny. "dBASE tutor part 5 — database maintenance." *(1984, November)* 25

Humphress, Danny. "dBASE tutor part 6 — building a foundation for application programs." *(1984, December)* 23

Humphress, Danny. "dBASE tutor part 7 - files for two." *(1985, January)* 47

Humphress, Danny. "dBASE tutor part 8 — the operational calculating file drawer." *(1985, February)* 19

Humphress, Danny. "dBASE tutor part 9 — of strings and things." *(1985, March)* 53

Humphress, Danny. "dBASE tutor part 10 — all systems functioning." *(1985, April)* 21

Humphress, Danny. "dBASE tutor part 11 — the dProgramming zone." *(1985, May)* 44

Humphress, Danny. "dBASE tutor part 12 — screen formats: better than plain vanilla." *(1985, June)* 21

Humphress, Danny. "A few good bars (cover charge only)." *(1984, August)* 22 —Bar graph drawing.

Humphress, Danny. "The hierarchal tree." *(1985, June)* 17 —Tutorial on MS-DOS's hierarchal file directories.

Humphress, Danny. "Mastering MS-DOS part 1." *(1984, May)* 8 —As applied on the Model 2000.

Humphress, Danny. "Mastering MS-DOS part 2." *(1984, June)* 8

Humphress, Danny. "Mastering MS-DOS part 3 — exploring MS-DOS commands." *(1984, July)* 27

Humphress, Danny. "Mastering MS-DOS part 4." *(1984, August)* 16

Humphress, Danny. "Mastering MS-DOS part 5 — Edlin function keys." *(1984, September)* 33

Humphress, Danny. "Mastering MS-DOS part 6 — more on Edlin." *(1984, October)* 24

Humphress, Danny. "Mastering MS-DOS part 7 — a final look at Edlin." *(1984, November)* 14

Humphress, Danny. "The perfect companion." *(1984, April)* 32 —Review of the Model 2000.

Humphress, Danny. "Plumbing for beginners." *(1984, December)* 30 —MS-DOS tutorial.

Humphress, Danny. "The revised reader." *(1984, May)* 25 —Updated bar code program. Bar code listing.★

Humphress, Danny. "Scanning the pages of PCM." *(1985, January)* 27 —Updated bar code program. Bar code listing.★

Humphress, Danny. "The Tandy 1000: Tandy's surprise machine." *(1985, February)* 53

Humphress, Danny. "Two triple threats for the 2000: a comparative review of Lotus 1-2-3 and Super-Calc." *(1984, September)* 42

Ireland, Nathaniel. "The portable vacation." *(1985, June)* 26 —Keep running records of vacation expenses. Bar code listing.★

Ireland, Nathaniel F. "A Model 100 calculator." *(1984, June)* 35 —Use it as a calculator. Bar code listing.★

Ireland, Nathaniel F. "Navloc. *(1984, June)* 30 — Determine latitude and longtitude.

Ireland, Nathaniel F. "Short-term memory booster!" *(1984, March)* 32 —Appointment calendar.

Kintsfather, David. "The portable computer as a social science research tool." *(1984, September)* 17 —Comments about statistics.

Klages, Jon P. "3-D for your LCD." *(1984, August)* 30 —3-D graphics. Bar code listing.★

Kleinfeld, Andrew J. "Dump it on Compuserve (and set yourself free)." *(1984, January)* 14 —Use Compuserve to hold your files.

Lord, Vincent, Jr. "Determine your return on investments." *(1985, March)* 10 —Bar code listing.★

Lord, Vincent, Jr. "Determining gross profit margins." *(1983, August)* 7

Lord, Vincent, Jr. "Financial ratios can help determine corporate 'soundness.'" *(1983, July)* 14

Lord, Vincent, Jr. "Let's go spelunking but watch out for the Wumpus." *(1983, October)* 29

Lord, Vincent, Jr. "Quick menu—a programming utility." *(1984, April)* 27 —Add menus to Basic programs. Bar code listing.★

Lord, Vincent, Jr. "Sailing the Mediterranean with PoCo." *(1983, October)* 16 —Use Model 100 for navigating.

Lord, Vincent, Jr. "Three methods of property depreciation." *(1984, January)* 28

Mendenhall, R. E. "Decisions, decisions—using your Model 100 as a decision making aid." *(1984, October)* 37

Mendenhall, R. E. "It's a date!" *(1985, May)* 53 —Calendar calculations. Bar code listing.★

Mitchell, D. E. "The PoCo-CoCo connection." *(1984, August)* 26 —Transfer files to the Color Computer.

Nolan, Bill. "Examining Basic's INSTR function." *(1984, May)* 35 —Basic tutorial.

Nolan, Bill. "PoCodometer." *(1984, October)* 41 —Speedometer check. Bar code listing.★

Oppedahl, Carl. "Give your Model 100 an auto-answer modem for under $10." *(1984, November)* 30

Oppedahl, Carl. "Going online at the office with PoCo." *(1984, July)* 22 —Direct connect to multi-line office phone.

Oppedahl, Carl. "How to send graphics characters to a dot matrix printer." *(1985, May)* 39 —Bar code listing.★

Paludan, Ronald. "Appending PoCo files." *(1983, November)* 24

Paludan, Ronald. "How to make your RAM files invisible." *(1984, February)* 29 —Files will not appear on the menu.

Paludan, Ronald. "The PoCo calendar." *(1983, December)* 18 —Print calendars on the screen.

Paludan, Ronald. "Try this triangle challenge." *(1983, September)* 18 —The game of SIM.

Perkins, Mel. "Meet Professor PoCo." *(1984, July)* 32 —Math drill. Bar code listing.★

Perkins, Mel. "Spelling practice." *(1984, February)* 11

Petersen, Cheryl. "Whose skies are user friendly?" *(1984, April)* 22 —More on portables in airplanes.

Pifer, David R. "Making pies a piece of cake." *(1984, September)* 11 —Pie charts on the 2000.

Pifer, David R. "Skedit - part 1." *(1984, December)* 8 —A full featured graphics editor for the Model 2000.

Pifer, David R. "Skedit - part 2." *(1985, January)* 15

Pifer, David R. "Skedit - part 3." *(1985, February)* 44

Pifer, David R. "Skedit - part 4." *(1985, March)* 34

Preble, Laurence D. "Get 'hands on' experience with this keyboard sender." *(1984, January)* 16 —Use Model 100 to send Morse Code.

Preble, Laurence D. "Seller-friendly." *(1984, December)* 40 —Order form and invoices. Bar code listing.★

Randall, Larry. "Automatic line numbering for Basic." *(1984, March)* 30

Randall, Larry. "Let your portable list those variables." *(1984, February)* 30 —Basic programming utility.

Randall, Larry. "Looking for ROM." *(1984, January)* 19 —Disassembler to look at the memory.

Randall, Larry. "A renumbering routine for PoCo." *(1983, December)* 32 —Renumber Basic programs.

Rideout, Ralph. "Translate plotting tedium into ease with Graphplot 2000." *(1984, October)* 12

Roberson, Larry M. "You can 'playit' on your PoCo." *(1984, September)* 21 —Bar code listing.★

Rogers, Richard T. "Snoop-proof your memory!" *(1983, December)* 33 —Add a password protection.

Rollins, Willis. "Examining your machine's capabilities." *(1984, February)* 32 —Tutorial on the Model 100.

Rosen, Bob. "Inside and outside." *(1983, August)* 22 —General overview of Model 100.

Rosen, Bob. "Telecommunications and the Model 100." *(1983, July)* 13

Rothstein, Rick. "PoCo and the bandit." *(1985, February)* 22 —Slot machine game.

Rothstein, Rick. "Screen dump 100." *(1985, January)* 43 —Graphics screen dump program. Bar code listing.★

Rothstein, Rick. "Square deal." *(1984, June)* 21 —Version of poker solitaire. Bar code listing.★

Sanders, Wayne. "Flag of the month club - Australia." *(1984, November)* 36

Sanders, Wayne. "Flag of the month club - Canada." *(1984, June)* 11

Sanders, Wayne. "Flag of the month club - Great Britain." *(1984, August)* 36

Sanders, Wayne. "Flag of the month club - South Korea." *(1984, September)* 36

Sanders, Wayne. "Flag of the month club- Olympic games." *(1984, July)* 37

Sanders, Wayne. "The gallery." *(1985, January)* 35 —Graphics demo.

Sanders, Wayne. "The gallery." *(1985, February)* 52

Sanders, Wayne. "The gallery." *(1985, March)* 12

Sanders, Wayne. "The gallery." *(1985, April)* 42

Sanders, Wayne. "The gallery." *(1985, May)* 31

Sanders, Wayne. "The gallery." *(1985, June)* 42

Sanders, Wayne. "Those great (and colorful) '2000' graphics!" *(1984, May)* 37 —Draw a U.S. flag.

Sanders, Wayne. "2000 gallery." *(1984, December)* 16 —Graphics demo.

Schwartz, Steven A. "The exchange factor." *(1985, April)* 49 —Foreign currency converter. Bar code listing.★

Sissala, Gary A. "Create letter perfect text files with this Print Format Program." *(1983, October)* 34 —Improves the TEXT output.

Socci, Vance. "DIALM8 for a handy autodialing extender." *(1983, November)* 19 —Improvements to ADRS program.

Stanford, Michael. "Drawing a pie chart on the screen." *(1983, August)* 24

Stauffer, William T. "Disk video interface: a user's critique." *(1984, May)* 17 —Includes how to install second drive.

Stauffer, William T. "Homemade PoCo to CCR cable." *(1983, October)* 32

Stauffer, William T. "Six financial programs for PoCo." *(1984, February)* 24 —Amortization, etc.

Stevens, Guerri F. "Download aid." *(1983, October)* 26

Trevor, Alexander. "Weather download—the low-down." *(1983, December)* 16 —Download the Vidtex radar maps from Compuserve.

Wheeler, Dale. "Among printer utilities LLST is without parallel." *(1983, September)* 21 —Utility to use serial interface to printer.

Wheeler, Dale. "Organize your ADRS.DO files with this 'label maker.'" *(1983, November)* 14 —(Correction, December, 1983, page 37)

Wheeler, Dale. "Pruning your programs." *(1984, January)* 26 —Basic utility

White, Richard A. "A Basic start." *(1983, October)* 6 —Tutorial on Basic.

White, Richard A. "The bottom line: settling up the expense account." *(1985, April)* 10 —Wrap-up of expense account program.

White, Richard A. "Dealing with Basic string functions." *(1984, August)* 8 —Tutorial on Basic.

White, Richard A. "Gaining sophistication in printing techniques." *(1984, March)* 8 —Tutorial on Basic.

White, Richard A. "Getting a line on your computer's operators." *(1984, September)* 27 —Tutorial on Basic.

White, Richard A. "Groundwork for future file handling." *(1984, April)* 9 —Tutorial on Basic.

White, Richard A. "Housekeeping for the expense account." *(1985, March)* 61

White, Richard A. "An introduction to spreadsheets." *(1985, June)* 10

White, Richard A. "Keeping the expense account." *(1985, January)* 50 —Tutorial with practical example.

White, Richard A. "More on variables." *(1984, July)* 34 —Tutorial on Basic.

White, Richard A. "Program control." *(1983, December)* 8 —Tutorial on Basic.

White, Richard A. "Program logic: a discussion in context." *(1984, October)* 33 —Tutorial on Basic. Bar code listing.★

White, Richard A. "Programming to save memory." *(1984, January)* 12 —Tutorial on Basic.

White, Richard A. "Setting up the expense account." *(1985, February)* 26

White, Richard A. "Some 'constants' about variables." *(1983, November)* 8 —Tutorial on Basic.

White, Richard A. "Techniques for handling data." *(1984, February)* 8 —Tutorial on Basic.

Zucconi, George. "Data and file transfer between the Model 100 and CP/M computers." *(1985, February)* 9

BAR CODE LISTINGS

Balonis, Ronald F. "Know the size of your files and programs at a glance." *(1985, May)* 33 —Bar code listing.★

Balonis, Ronald F. "Test your intuitive decision making skills." *(1985, April)* 53 —Decision making program. Bar code listing.★

Cornman, Aileen ; and Cornman, John. "The PoCo word processor." *(1985, February)* 37 —Word processing program. Bar code listing.★

Craven, Richard. "Logging account time with 'Time Clock.'" *(1984, September)* 25 —Bar code listing.★ Correction, January 1985, page 62.

Delbourgo, Bob. "Keyboard Dominoes." *(1984, November)* 11 —Bar code listing.★

Frowenfeld, Robert. "Expense reporting in search of the 'round toit.'" *(1984, May)* 15 —Expense monitor program. Bar code listing.★

Frowenfeld, Robert. "A graph is worth a thousand digits." *(1984, July)* 8 —How to draw bar graphs. Bar code listing.★

Guscott, J. Kenneth. "The executive decoder ring: a different approach to file security." *(1984, August)* 11 —Put your files in code. Bar code listing.★

Guscott, J. Kenneth. "Five easy pieces for PoCo." *(1984, May)* 20 —A music program. Bar code listing.★

Humphress, Danny. "Bar code Basic." *(1984, April)* 17 —Bar code reading program. Bar code listing.★

Humphress, Danny. "The revised reader." *(1984, May)* 25 —Updated bar code program. Bar code listing.★

Humphress, Danny. "Scanning the pages of PCM." *(1985, January)* 27 —Updated bar code program. Bar code listing.★

Ireland, Nathaniel. "The portable vacation." *(1985, June)* 26 —Keep running records of vacation expenses. Bar code listing.★

Ireland, Nathaniel F. "A Model 100 calculator." *(1984, June)* 35 —Use it as a calculator. Bar code listing.★

Klages, Jon P. "3-D for your LCD." *(1984, August)* 30 —3-D graphics. Bar code listing.★

Lord, Vincent, Jr. "Determine your return on investments." *(1985, March)* 10 —Bar code listing.★

Lord, Vincent, Jr. "Quick menu—a programming utility." *(1984, April)* 27 —Add menus to Basic programs. Bar code listing.★

Mendenhall, R. E. "It's a date!" *(1985, May)* 53 —Calendar calculations. Bar code listing.★

Nolan, Bill. "PoCodometer." *(1984, October)* 41 —Speedometer check. Bar code listing.★

Oppedahl, Carl. "How to send graphics characters to a dot matrix printer." *(1985, May)* 39 —Bar code listing.★

Perkins, Mel. "Meet Professor PoCo." *(1984, July)* 32 —Math drill. Bar code listing.★

Preble, Laurence D. "Seller-friendly." *(1984, December)* 40 —Order form and invoices. Bar code listing.★

Roberson, Larry M. "You can 'playit' on your PoCo." *(1984, September)* 21 —Bar code listing.★

Rothstein, Rick. "Screen dump 100." *(1985, January)* 43 —Graphics screen dump program. Bar code listing.★

Rothstein, Rick. "Square deal." *(1984, June)* 21 —Version of poker solitaire. Bar code listing.★

Schwartz, Steven A. "The exchange factor." *(1985, April)* 49 —Foreign currency converter. Bar code listing.★

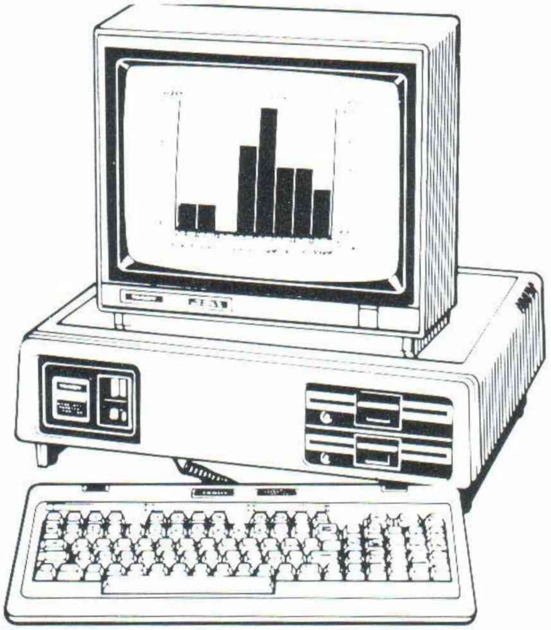
White, Richard A. "Program logic: a discussion in context." *(1984, October)* 33 —Tutorial on Basic. Bar code listing.★

Leslie A. Foster is a librarian with Dalhousie Ocean Studies Programme, Dalhousie University, Halifax, Nova Scotia, Canada. He is the co-editor of Marine Affairs Bibliography, an index to the law of the sea literature.

NOCONA

ELECTRONICS

TRS-80 Model 2000 by Tandy®



MS-DOS Based

Computer System

256K MOD2000 W/MONO MONITOR	1319.00
256K MOD2000 W/COLOR MONITOR	1649.00
256K MOD 1200 HD 2 DR.	1499.00
128K MOD 1000 W/MONO 1 DR.	839.00
128K MOD 1000 W/COLOR 2 DR.	1199.00
24K MOD 100 PORT	399.00
24K MOD 200 PORT	769.00
64K MOD 4 2 DR.	925.00
64K MODEL 4P 2DR.	935.00
512K MOD 6000 2DR	3099.00
80K MOD 12 2 DR.	1199.00
15 MEG HARD DISK	1329.00
35 MEG HARD DISK	2059.00
DMP 105 PRINTER	145.00
DMP 110 PRINTER	269.00
DMP 120 PRINTER	349.00
DMP 430 PRINTER	585.00
DMP 2100P PRINTER	1019.00
CGP 220 COLOR PRINTER	499.00
DWP 510 PRINTER	999.00
TRACTOR DWP 210	99.00
TRACTOR DWP II/510	159.00
TRACTOR FEED DMP2100P	99.00
PARALLEL PRINTER SWITCH	96.00
PARALLEL PRINTER CONTROLLER	179.00
PARALLEL PRINTER CABLE	29.00
DISK VIDEO INTERFACE	399.00

TRS-80® Model 100 by Radio Shack

100% RS COMPONENTS NO FOREIGN DRIVES OR MEMORY FULL WARRANTY
ALL RS SOFTWARE 20% OFF CATALOG PRICE
CASHIERS CHECK OR MONEY ORDER MUST ACCOMPANY ALL ORDERS

(817) 825-4027

NOCONA ELECTRONICS • BOX 593 • NOCONA, TX 76255

Follow the Yellow Brick 'Path'

By Danny Humphress
PCM Technical Editor

In the June DOSsier, we began our explorations through labyrinth of the MS-DOS hierarchal directory system. We continue our adventure this month as we discover special new commands designed to work with directories.

If you were with us on our most recent tour of MS-DOS, you will recall that MS-DOS versions 2.00 and higher allow you to create special files called *directories* which may contain other files and even other directories. The MS-DOS commands `MKDIR` and `CHDIR` allow you to create new directories and move from one to another.

Killing Directories

As discussed earlier, a directory is simply a file which contains other files. The similarity to a simple file ends here, though. When you use MS-DOS commands such as `COPY` and `ERASE`, the files *within* the directory are affected — not the directory itself. If you have a directory called `AR` and you enter the command `ERASE AR`, the *directory* would not be erased, but the *files* within that directory would be. Be careful. If you had other directories within `AR`, you would not be allowed to erase it at all until you removed those directories.

This brings us to a logical question: How do you remove directories? Not with `ERASE`! A special command, `RMDIR` (may be abbreviated `RD`), will remove a directory.

Dead Directories Don't Have Files

As a safety feature, `RMDIR` will only remove an empty directory. You must first use `ERASE` to remove all the files from the directory, and make sure that there are no directories below it. If there are other directories below the one you want to delete, you will have to go to the lowest directory and begin your destruction there. Once again, you may not remove a directory until it's completely empty.

Another `RMDIR` safety feature prevents you from ripping out the floor from beneath your feet. In other words, you cannot remove a directory that is your current default directory. To use `RMDIR`, you must move "up" at least one level (toward the root directory) from the directory you are removing.

Since you cannot go higher than the root directory, it is impossible to remove it. On the other hand, it *is* possible to remove the root directory using another method — erase the disk!

Now, let's remove our fictitious `AR` directory. We'll suppose that `AR` is a subdirectory of the root directory, that `AR` is our current default directory, and that there are no directories within `AR`. These commands would get the job done:

`CHDIR \`	Move to the root directory
`ERASE AR`	Erase all files within `AR`
`RMDIR AR`	Remove the directory

PATH

They say, "build a better program and the world will beat a path to your directory."

Well, I may have taken a *few* liberties with that phrase, but only enough to make it serve my purpose. The problem with hierarchal directories, you see, is that one never seems to be in the same directory as the program you want to run. MS-DOS' "external" command programs such as `CHKDSK`, `FORMAT`, and `DISKCOPY` are usually kept in the root directory. If you were in a lower directory and wanted to use one of these programs, you would normally have to move to their directory.

There *is* another way. The `PATH` command allows you to give MS-DOS a list of places to look when it can't find a program in the current directory. Once you've given it the list, it will remember it until you change it or reset the computer. In the above case, you would have wanted to tell MS-DOS to look in the root directory after searching the current directory.

For our above example, the `PATH` command like would be:

```
PATH \
```

This tells MS-DOS to search the root directory (\ means "root") if it fails to find a program in the current directory.

As mentioned, you may specify an entire list of paths. MS-DOS will look in each one of them until it finds the program you're trying to run. To do so, you just separate the paths with semicolons (`;`). Example:

```
PATH \;\UTILITY
```

The above command would have MS-DOS search the root directory and then a directory called UTILITY if it could not find the program in the current directory.

Remember, you only have to use the PATH command once. MS-DOS will remember it throughout the entire session with the computer. Once you've come up with the PATH sequence that best serves your situation, it's a good idea to make it a part of your AUTOEXEC.BAT batch file so that it is automatically set up each time you reset the computer (with that particular disk).

Limitations

You may have noticed that I used the words "command" and "program" a lot while talking about the PATH command. That is because they are the only types of files that PATH will work with. PATH will cause MS-DOS to search for "executable" (machine language) programs that have either a .COM or .EXE extension.

As a rule of thumb, unless you want to run a program that is executed by just typing the name at the MS-DOS prompt, PATH will not work for you. Likewise, most programs are not "smart" enough to use the path list you specified when they are looking for other programs and/or files. They will generally expect the files to be in the current directory.

Practical Paths

MS-DOS external commands (those that are actually separate programs) such as CHKDSK, EDLIN, FORMAT, DISKCOPY, etc., all work well with PATH. If you have a Tandy 1000 or 2000, you may use path to point to the directory that contains BASIC and be able to get into that programming language from any directory. Since BASIC on the Tandy 1200 is in two separate programs, the first program will not be able to find the second part unless it's in the current directory.

Dot and Doubledot

You may have noticed that when you did a DIR command on a subdirectory, there were two odd entries at the beginning of the list. The two directories '.' and '..' within each subdirectory are not *real* directories, but rather they are *pointers* to directories.

The single period '.' points to the current directory and may be used within any command to specify the *current* directory. The really useful "phantom entry" is the double period '..'. It is used to specify the *previous* directory (the parent of the current directory).

Therefore, the command DIR . would give you a listing of the current directory, and DIR .. would give you a listing of the previous directory.

If you wanted to move up one directory but could not remember its name, CHDIR .. would take you there. Another CHDIR .. would move you up yet another level, and so on until you reach the root directory.

Once again, '.' and '..' may be used anywhere you would use full path names.

Onward!

With an understanding of hierarchal directory systems firmly seated in your mind, we are ready to explore deeper into MS-DOSdom. Next month, we'll search for the true meaning of device drivers. You know device drivers — those funny .SYS files that came on your MS-DOS disk. You'll know what they're for after you read the next DOSsier. PCM

MS-Tips

Hierarchal directories give you the power to categorically organize the programs and files on a disk. With the massive storage of hard disk drives, it would be a Herculean task to search through a long, unorganized, directory listing to find one file in the "haystack." Organizing files in separate directories is the logical solution, but as the directories themselves multiply in number, it becomes hard to remember their names.

MS-DOS does not provide a built-in method for getting a list of directories, but it *can* be accomplished with this command line:

```
DIR *.¦FIND“>”
```

Since the command is not very easy to remember, nor is it the simplest to type, it is a good candidate to be made a part of a batch file. A batch file named LISTDIR would be appropriate.

This command line takes advantage of the fact that the only files that have the greater-than character ('>') in their directory entry are directories. Directories have <DIR> after their name in the list.

DIR *. lists the files that have no extension. They are then piped through the FIND filter and only those lines with a '>' character are printed. Using the '*.' wildcard is not necessary, but it narrows down the number of lines that have to be passed through the FIND filter, thus making the process a bit faster.

When going to do battle with the automobile sales force, arm yourself with the facts. Take along a printout from . . .

The Car Shopper's Spreadsheet

By Richard A. White
PCM Contributing Editor

"Bigger is better only when it is shown that smaller just won't do the job."

What's buying a car got to do with spreadsheets or other computer programming? A lot, if you use the spreadsheet in the process. I just went through the process. I knew pretty much what I wanted so I didn't do many "what-if" projections. If you feel like looking at a variety of makes and models, more spreadsheet use could be involved.

Any spreadsheet will do. There are at least a dozen that will run on a Tandy 1000 or 1200 starting with the one that kicked it all off, *VisiCalc*. Later spreadsheets, including now champion *Lotus 1-2-3*, basically conform to conventions established by *VisiCalc*, though with many enhancements. Since my experience is mostly with *Lotus* and other *VisiCalc* type sheets, I will use examples of that type for the most part. *DeskMate*, *Multiplan* and others do differ in significant ways. But when you understand what is being done, you should be able to interpret how to do it in your particular sheet.

If you have not used a spreadsheet before and are not sure what you will use one for, don't spend big dollars for one with all the bells and whistles at first. Tandy 1000 owners are particularly fortunate since *DeskMate* provides all the spreadsheet that 90 percent of us would ever need. You will do a lot better with a program that is friendly and easily learned than with a big one that is somewhat intimidating with functions you will never use. Bigger is better only when it is shown that smaller just won't do the job.

Basically, you really can get organized easily using a spreadsheet. And when you are organized, you can be on the offense with the car dealer on the defense or even make it a team effort. Let's start the story.

First, I got $500 free money towards a new car. Some may remember that when Chrysler announced record earnings in January, they also announced that all employees and all customers who had purchased a new Chrysler product between 1979 and 1984 would get a certificate worth $500 on a new 1985 car. My Iacocca Certificate arrived within a week.

(Richard White has a long background with microcomputers and specializes in BASIC *programming. With Don Dollberg, he is the author of the* TIMS *database management program for the Color Computer.)*

One must address these situations with great care. After all, the '72 battleship . . . excuse me, station wagon was still getting from gas station to gas station and my '80 Horizon that won the certificate is still new by comparison. It's not that I am tight. There are three computers in the house. The fact that two are CoCos and one is a Model 100 says only that I look for best value for the buck.

I reflected for a couple of weeks. The battleship was aging, rusting and pieces were falling off. Signals of middle age perhaps. But, a Tandy 1000 seemed a bit neater than a new car and is unlikely to rust. On the other hand, four computers would never fly. It seemed that a brief fact-finding trip to the local Dodge dealer would be in order. A new wagon or perhaps a Caravan might be considered.

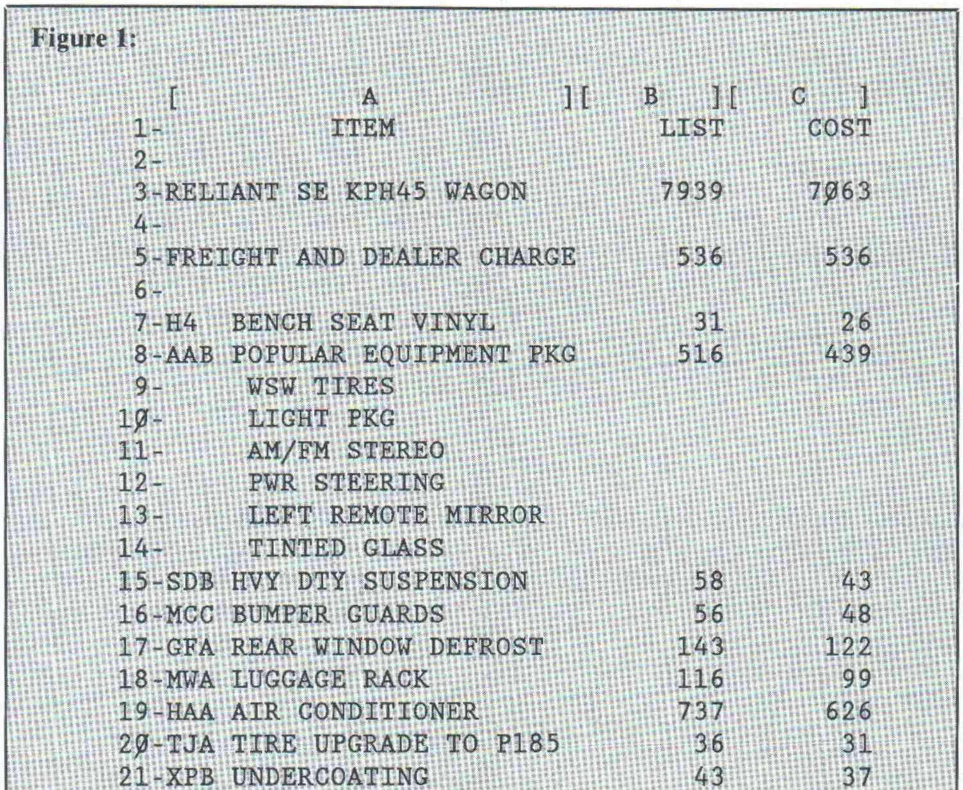
The dealer was out of wagons and had only one Caravan that cost more than four IBM PCs with a printer thrown in. I did get literature and stopped at a newsstand on the way home to buy a book containing both suggested retail and dealer costs for new cars and their options. Now I could rationally evaluate the situation from the safety of my easy chair with my wallet secured between me and the upholstery.

On the way home the next evening, I stopped at a Plymouth dealer who had no new wagons either. He would sell me an '84 demonstrator where my Iacocca Certificate did not apply and which did not benefit from the free automatic transmission promotion. I had the feeling I would pay more for that car than a new one ordered to my specifications. Time to do a spreadsheet.

Figure 1 is the final spreadsheet for the car I ordered and is typical of what you might do a number of times as you narrow down your choices. All the data comes right out of the manufacturer's literature and the price book. The only formulae, @SUM(B1 . . . B23) and @SUM(C1 . . . C23), are in cells B24 and C24, which sum the entries in the columns above the total row including the heading and the blank row above the totals. You can insert and delete any row within this range and the formulae will adjust themselves. This makes "what-if" testing easy.

For example, you may have chosen three options that are also included in the discounted popular options package. Save your current spreadsheet. Then delete those options and add the popular options package. A recalculation shows the new totals which you can compare with those in the unmodified spreadsheet. Since the popular options package is discounted, you might find that $50 more buys $150 worth of options.

Figure 1:



```
      [             A             ][    B    ][    C    ]
      1-           ITEM                 LIST       COST
      2-
      3-RELIANT SE KPH45 WAGON          7939       7Ø63
      4-
      5-FREIGHT AND DEALER CHARGE        536        536
      6-
      7-H4  BENCH SEAT VINYL              31         26
      8-AAB POPULAR EQUIPMENT PKG        516        439
      9-      WSW TIRES
     1Ø-      LIGHT PKG
     11-      AM/FM STEREO
     12-      PWR STEERING
     13-      LEFT REMOTE MIRROR
     14-      TINTED GLASS
     15-SDB HVY DTY SUSPENSION            58         43
     16-MCC BUMPER GUARDS                 56         48
     17-GFA REAR WINDOW DEFROST          143        122
     18-MWA LUGGAGE RACK                 116         99
     19-HAA AIR CONDITIONER              737        626
     2Ø-TJA TIRE UPGRADE TO P185          36         31
     21-XPB UNDERCOATING                  43         37
     22-NHM AUTO SPEED CONTROL           179        152
     23-
     24-                  TOTAL        1Ø39Ø       9222
```

The one number you won't be able to exactly define without visiting the dealer is the freight and dealer charge. If you do a lot of looking and asking, you will pick up some examples of this and have a ballpark number to apply to various cars.

Those with printers will do well to print each sheet they make and take these along when they visit the dealers. On my next dealer visit, I had done some spreadsheets, but did not have them along. Having done the spreadsheets only the night before, I knew from memory pretty well what I wanted and the costs to the dealer for these.

Now this dealer had just finished the previous month as top in the district and was on a roll. Sensing I knew what I wanted, the sales person got a scrap of paper, pencil and her price book and set about recreating my spreadsheet by hand. After minutes of writing, erasing, adding and subtracting, we had a list number to work from. I was thinking about how much easier the same thing had been with my spreadsheet.

Now came haggling time. I knew delivery times and that my cost numbers were right. We talked about what price might fly and I got up to leave. After all, I never buy a car without completely checking out the competition and I still wanted to look at a few different options. That got some action and a $200 over cost offer.

The next day, we arrived with the above spreadsheet to finalize the order. There were still the color questions to decide. Again pencils, papers and calculators came out as they strove to determine the price. And there were a few suggestions. "We recommend tinted glass with air conditioning."

After a brief glance at the printout, I said, "It's included in the popular options package."

"Oh."

The first cut price was wrong. They left out the vinyl seats. Finally they got it right. I should have made two printouts and given them one to check. It would have saved some hassle and mistakes.

Next came the financing. I looked in a few references for the formula for calculating monthly payments and came up dry when Herb Slodounik of

Text processing power that no other program can equal.

WriteROM©

on Snap-in™ Cartridge $149.95

PCSG says "Send it back in 30 days for a full refund if you don't agree."

WRITE ROM is the definitive word processing extension for the Model 100. Less than two months after the Model 100 was announced Portable Computer Support Group introduced the very first text formatter for the Model 100. That program, called Write+ was licensed to Tandy and is now in Radio Shack Computer Centers as Scripsit-100. Write+ had many powerful features and most reviewers still say it is the best of the cassette based text formatters.

But now eighteen months later PCSG has introduced WRITE ROM. Those who experience it have said "WRITE ROM literally doubles the text processing power of the Model 100."

WRITE ROM is what you would have expected PCSG, the software leader for the Model 100, to develop in the 18 months since Write+ was brought to the market.

First of all WRITE ROM as it's name implies is on a snap in ROM. You simply take a quarter and open the little compartment on the back of your Model 100 and press in the ROM cartridge. It is as easy as an Atari game cartridge and can be snapped in and out instantly so that you can use other ROM programs whenever you wish.

WRITE ROM appears on the main menu just like one of your built in programs. It lets you do every formatting function you would expect like setting margins, centering, right justifying and having headers and footers. But it does them under function key control, with the clear and easy to learn and use techniques for which PCSG has become famous.

In keeping with PCSG's long standing reputation for superlatively simple yet comprehensive documentation, the manual is a model of lucidity.

WRITE ROM remembers your favorite format settings so that you can print a document without any set up, but you can change any formatting or printing parameters instantly with a function key.

WRITE ROM's 'pixel mapping' feature shows you an instant picture on the screen of how your printout will look on paper. Incidently, PCSG introduced this feature on the Olivetti M-10 version of Write+ over a year ago.

In all there are 44 separate features and functions that you can do with WRITE ROM, and some of these features are truly breakthroughs for the Model 100.

First, WRITE ROM lets you do search and replace, with function key ease of course. Any word or phrase in a document can be searched for and replaced with any other phrase where the search words appear.

Second, WRITE ROM lets you send any text (formatted or not) to any other computer over the phone with just a function key. What's more, it dials and handles sign on protocol automatically.

Third, WRITE ROM has a wonderful feature called 'Library' that gives your Model 100 power that you never thought it could have. Library lets you record favorite phrases, words, or commonly used expressions (sometimes called boilerplate). Any place you wish any library text to appear in your document you just type in a code. WRITE ROM automatically inserts the text just like a Xerox Memory Writer.

The library phrase is inserted as your document is being printed rather than as it is being typed, so this feature conserves memory in documents where a long phrase is used repetitively, since each occurrence of a library phrase in your document is indicated by a single code character.

This Library feature is so powerful these two pages could be devoted just to telling you about things it can do. For example, you can have names and addresses that you designate in one text file with a customer or supplier number. Or you can have inventory items with stock numbers.

In your document you simply type in the customer or stock number and that entry from the other file is automatically inserted in the document. Picture what you can do with that kind of capability.

Because WRITE ROM is written in machine code, it is blindingly fast. No one can claim faster operation.

Because it is on a ROM it uses virtually none of your precious RAM for it's operation, and it does not interfere with other machine code programs in your RAM. It works with any printer, serial or parallel. At the touch of a function key you can find the size of a RAM file in bytes and in words (ideal for journalists and other writers who need to know how many words are in a piece). You can make a duplicate copy of a document file under a new name. You also can rename or delete (kill) any RAM file with function key ease.

This description only scratches the surface of this amazingly powerful piece of software. You can automatically insert the date or the time anywhere in your document; WRITE ROM senses when you are nearing the bottom of a page, and at your command will start a new paragraph on the next page.

Write + was the Model 100 pioneer in the use of 'dot commands' to allow control of such things as margins, centering, line spacing and other appearance related changes in the middle of a document. WRITE ROM goes a step further by making all the dot commands Wordstar compatible. This means that if you wish you can quite easily prepare a Wordstar compatible document. Then you can use features of WRITE ROM (such as pixel mapping) that Wordstar lacks, before uploading to your desktop.

A Mail Merge feature allows you to send the same document to every name on your mailing list, personalized for each recipient.

WRITE ROM enables you to do underlining, boldface and correspondence mode as well as any other font feature that your printer supports in a way that is so unique many users say "It is worth the price of the program just to have this one feature."

Here's how it works: When you want to underline you don't have to remember some complicated printer code. You just type Graph U, and to end underline you just type Graph U again. For boldface it's Graph B and to end boldface it's Graph B again. It's easy to remember and easy to do. WRITE ROM lets you record the codes from your printer's manual one time only and then just use these easy to remember signals any time you want to do a printer font feature.

WRITE ROM does so many things that other text formatters cannot do.

For example you can not only double space but triple, quadruple or any other.

WRITE ROM allows you to use your TAB key in a doument so that you can indent the first line for a paragraph easily or space rapidly over many tab stops.

WRITE ROM has another nice feature. It allows you to undent. This means that you can have paragraphs that have a first line that projects to the left of the remainder of the paragraph.

WRITE ROM allows you to not only center a word or phrase on a line but you can center copy vertically on a page as well.

There are many other examples of excellent programming evident in WRITE ROM. The line feed problem of the Model 100 is dealt with by the simple use of a function key. Files are selected by moving the wide bar cursor over the WRITE ROM menu.

PCSG makes the claim that WRITE ROM is the easiest, fastest and most feature rich text formatter for the Model 100, as well as being the only one on a Snap-in ROM. You can do more with WRITE ROM than anyone thought possible for the Model 100. We at PCSG are happy to offer WRITE ROM because it expands the Model 100 to a dimension of text processing you cannot equal on even larger computers.

If you are already a PCSG customer you know the impressive quality of PCSG craftsmanship. We brashly state that WRITE ROM is the best you can buy. But don't take our word for it. It is sold on a thirty day trial. If you aren't as excited as we are, return it within 30 days for a full refund. Priced at $149.95, on Snap-in ROM. Mastercard, Visa or COD.

First with
software for
the Model 100

PORTABLE
COMPUTER
SUPPORT
GROUP

1-214-351-0564
11035 HARRY HINES BLVD. No. 207
DALLAS, TEXAS 75229

PCSG provides hotline software support for Model-100. Call 1-214-351-0564. Available now directly from Portable Computer Support Group. We endeavor to continue as The Leader in Software for the Model 100.

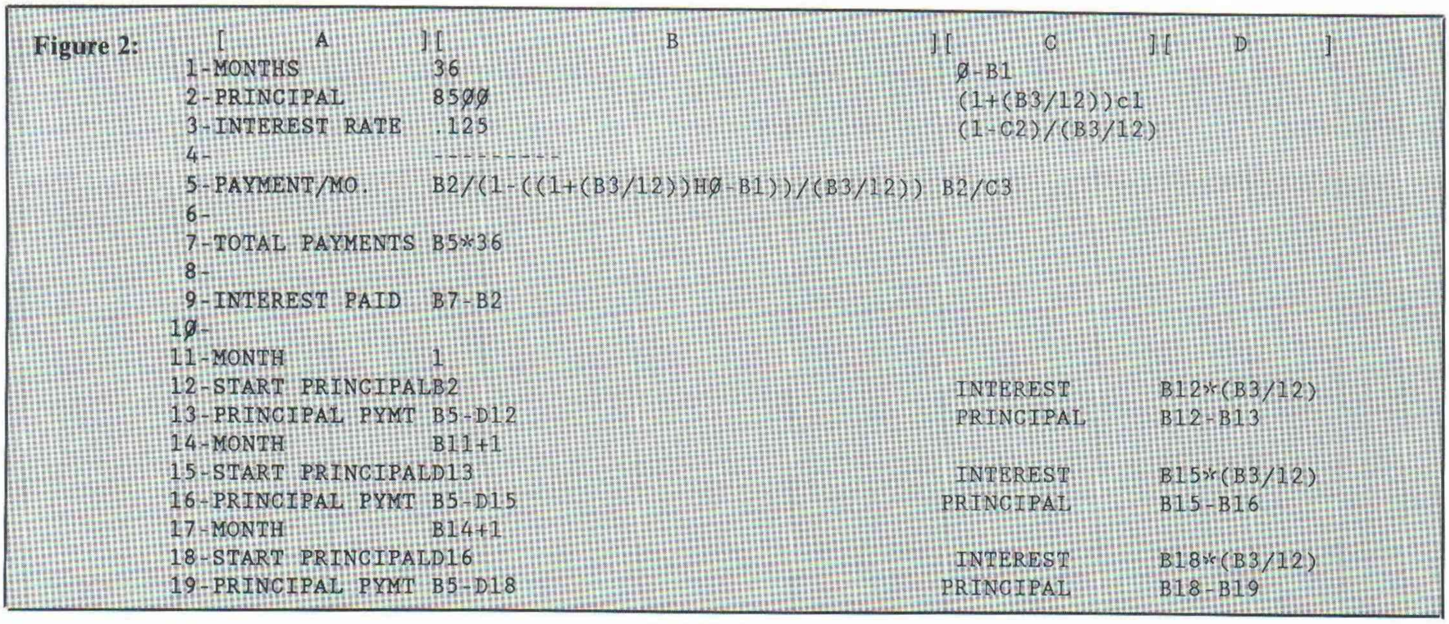


Figure 2:

```
   [      A      ][                 B                ][      C    ][      D     ]
 1-MONTHS         36                                  Ø-B1
 2-PRINCIPAL      85ØØ                                (1+(B3/12))c1
 3-INTEREST RATE  .125                                (1-C2)/(B3/12)
 4-               ---------
 5-PAYMENT/MO.    B2/(1-((1+(B3/12))HØ-B1))/(B3/12)) B2/C3
 6-
 7-TOTAL PAYMENTS B5*36
 8-
 9-INTEREST PAID  B7-B2
1Ø-
11-MONTH          1
12-START PRINCIPALB2                                  INTEREST      B12*(B3/12)
13-PRINCIPAL PYMT B5-D12                              PRINCIPAL     B12-B13
14-MONTH          B11+1
15-START PRINCIPALD13                                 INTEREST      B15*(B3/12)
16-PRINCIPAL PYMT B5-D15                             PRINCIPAL      B15-B16
17-MONTH          B14+1
18-START PRINCIPALD16                                 INTEREST      B18*(B3/12)
19-PRINCIPAL PYMT B5-D18                             PRINCIPAL      B18-B19
```

Decatur, Ill. sent me a letter with the formula saying it works in *Elite*Calc*, but not in *DynaCalc*. These are CoCo spreadsheets, but the problem exists in the PC compatible spreadsheets as well. The book formula is P=B((1–(1+(I/12))^–N)/(I/12)). BASIC, *Elite*Calc* for the CoCo and *Multiplan* for the Model 100 have a hierarchy of operators.

In our equation, these "languages" would raise (1+(I/12)) to the power –N and then subtract that value from 1. That is because raising to a power is above subtraction in the hierarchy. *VisiCalc* type spreadsheets do not have such a hierarchy. Therefore, *VisiCalc* and perhaps others evaluate all math expressions on a strictly left to right basis except as modified by parentheses. This is extremely important since it is contrary to how you would solve equations using pencil and paper or how you would program the equation in BASIC.

Figure 2 is a formula dump of a *VisiCalc* type spreadsheet to calculate monthly payments, given number of payments, amount financed (principal) and the interest rate given as a decimal number. An interest rate of 12.5 percent is entered as .125. These variables are entered in cells B1 . . . B3. The formula to calculate payments is in B5 and looks like this:

```
B2/(1-((1+(B3/12))HØ-B1))/(B3/12))
          (--1--)  (--3--)  (--5--)
       (-----2--)
      (---------4--------)
   (-----------------6-------------)
```

The diagram below the equation shows the order of calculation and how the parentheses group. The innermost expression, (B3/12) is calculated first. (1+Value1) and (0–B1) are at the same level so the left one is calculated first. In step four the raise-to-power operation occurs. At this point you should be able to reason out steps five through seven for yourself.

To get to the equation in B5, I broke the book formula into pieces and put these into cells C1 . . . C5. These produce the same answer in C5 as the full equation produces in B5. Many times it is easier to put work areas like C1 . . . C5 outside the formal area of the spreadsheet and move their results back to where that value is to appear in a printout. Now the consideration is the order of calculation for the whole spreadsheet rather than order of calculation for a specific formula.

A forward reference is where a formula in a cell uses a value from another cell that has not been calculated yet. A typical order is to calculate the values in Column A starting at Row 1. Then the spreadsheet moves to column B and does the same thing. This would work fine for the top part of our spreadsheet.

The last nine lines of the spreadsheet start a series of calculations that calculate how much of each payment goes to interest and to principal and then calculates the outstanding principal after each monthly payment. In cell B12, the starting principal is brought down from cell B2. This value is used in cell D12 along with the interest rate from B5. The interest charged in D12 is subtracted from the monthly payment in cell B5 to get the amount available to reduce the principal. In cell D13, the the starting principal for the month is reduced by the value from B13.

This spreadsheet arrangement becomes a disaster when columns are calculated from left to right. The calculation of the principal payment in B13 needs the interest payment from D12 which has not been calculated yet. The effect snowballs since the start principal in B15 needs the value from D13 which is not right because B13 is not yet right, etc.

You can manually cause a recalculation which will fix B13 and D13, but D13 is calculated too late for B15 to be right. In fact two recalculations for each month in the spreadsheet would be necessary. One of the beauties of *Lotus 1-2-3* is that it works out these forward references for you as does *Multiplan*. In other spreadsheets, you may not be so lucky, so check that your answers are right. Sometimes you may need to recalculate a few times extra.

Another type of calculation glitch is the circular reference.

```
 [    1    ][    2    ]
1[B1          A1
```

Cell B1 contains nothing but the formula A1. Cell A1 evaluates to 0 since there is nothing in B1. Likewise B1 equals 0 since A1 equals 0. Here no harm is done nor anything useful done, but in more complex expressions there is no telling what may happen. And, in those spreadsheets that do try to solve forward references, an error is announced. *Multiplan* proclaims "Circular references unresolved."

For those who have hung in there up to now, here is the interest payment spreadsheet as you would see it without the formula dump. See Figure 3.

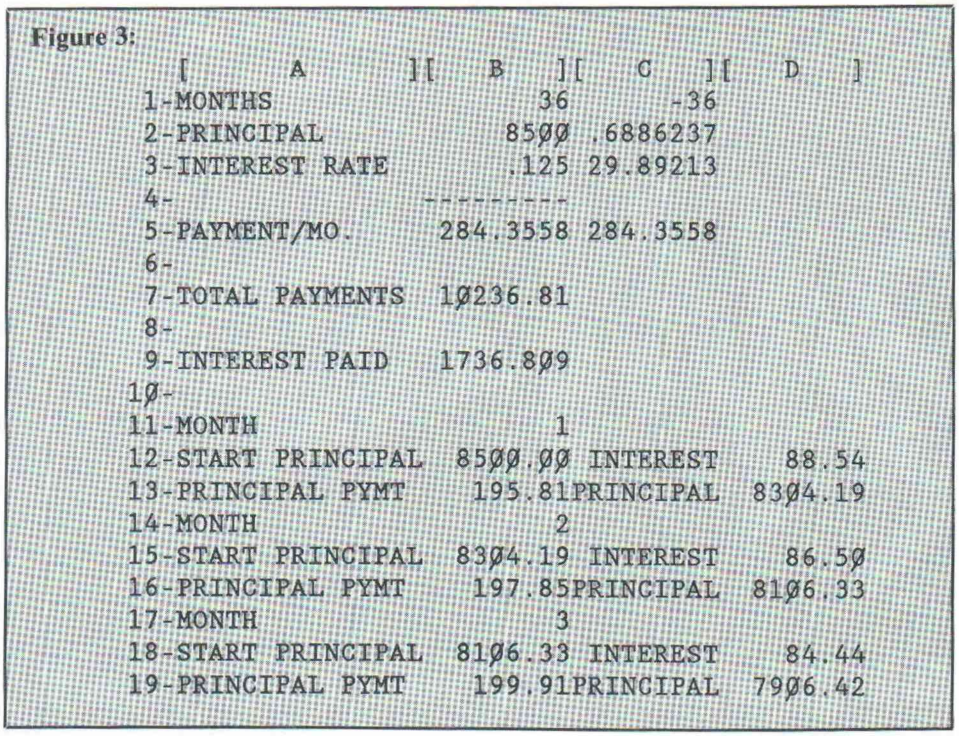


Figure 3:

```
        [       A       ][   B    ][   C    ][   D    ]
         1-MONTHS                36      -36
         2-PRINCIPAL           85ØØ .6886237
         3-INTEREST RATE       .125 29.89213
         4-               ---------
         5-PAYMENT/MO.     284.3558 284.3558
         6-
         7-TOTAL PAYMENTS  1Ø236.81
         8-
         9-INTEREST PAID   1736.8Ø9
        1Ø-
        11-MONTH                  1
        12-START PRINCIPAL  85ØØ.ØØ INTEREST      88.54
        13-PRINCIPAL PYMT    195.81PRINCIPAL   83Ø4.19
        14-MONTH                  2
        15-START PRINCIPAL  83Ø4.19 INTEREST      86.5Ø
        16-PRINCIPAL PYMT    197.85PRINCIPAL   81Ø6.33
        17-MONTH                  3
        18-START PRINCIPAL  81Ø6.33 INTEREST      84.44
        19-PRINCIPAL PYMT    199.91PRINCIPAL   79Ø6.42
```

I have shown only three months of the calculations of monthly principal and interest. The formulae for Month 1 are a bit different from those for succeeding months. The month number in B11 is entered as 1. The principal in B12 is obtained from B2. The month number for Month 2, B14, is calculated by adding 1 to the value in B11. The principal in B15 is the amount calculated in D13. The same pattern occurs for Month 3, B17 adds 1 to the value 3 cells above. The value in B18 is from the cell two up and two right. This is describing relative cell addresses. How many cells away in each direction is the cell from which to get a value?

Another type of cell addressing is fixed addressing. The interest rate is in B3. Whenever we need the interest rate we always go to that cell. Likewise, the monthly payment is in B5. We always go to that cell for it.

Now if we replicate or copy cells with formulae to enlarge our spreadsheet so it will produce interest and principal for the full term of the loan, some spreadsheets will display the formula in each cell and move the cursor to the first cell reference and ask (S)ame or (R)elative. If we want fixed addressing, we press the 'S' for same cell displayed. More likely than not we will be dealing with relative addressing and strike the 'R'.

In our example above, the payment amount and interest come from fixed address cells and we enter 'S' for them when asked. Two data pieces, the last month number and the outstanding principal, come from the previous month and relative addressing is required to get them. Finally, the amount of interest and the principal payment for the particular month are relatively addressed within the lines for that month.

When the copy is complete, the actual cell numbers are shown in the equations. My spreadsheet used the (S)ame or (R)elative information I entered to calculate the right cell address to use. This method is essentially the *VisiCalc* system. *Lotus 1-2-3* modified the method and made it fairly inscrutable. Someday I am going to sit down with a book and understand this part of *Lotus*.

If your head is swimming with fixed and relative addressing, let's try another example. I did the top nine lines of the spreadsheet in *Multiplan*. It currently resides in my M100. The formulae look like the example in Figure 4.

The formula in R5C2 is fixed addressing. It refers to cells specified by name. The formulae in R7C2 and R9C2 are relative addressing. R[-6] means the cell six rows above. The C means this column. Count them up and it's 36*(calculated payment). The interest is the total payment minus the principal.

Of course, you noticed that *Multiplan* is different in the way it designates columns and rows in expressions. This takes some getting used to if you migrate from another spreadsheet. I know people who have fallen in love with this and other aspects of *Multiplan*. It is more limited and therefore simpler than many others.

If I had had *Multiplan* in my M100 at the time, I could have done the car purchase analysis in it. I would not have been able to have the lengthy option descriptions and their list and cost figures on one line, so some abbreviations would have been used. Still the work would have been done. In fact, as final numbers such as freight and dealer prep became available, I could have entered them and had a recalculation on the spot. The people at the dealer could have copied these right off the machine. Ah, well . . . perhaps you will have a chance to play "impress the dealer."

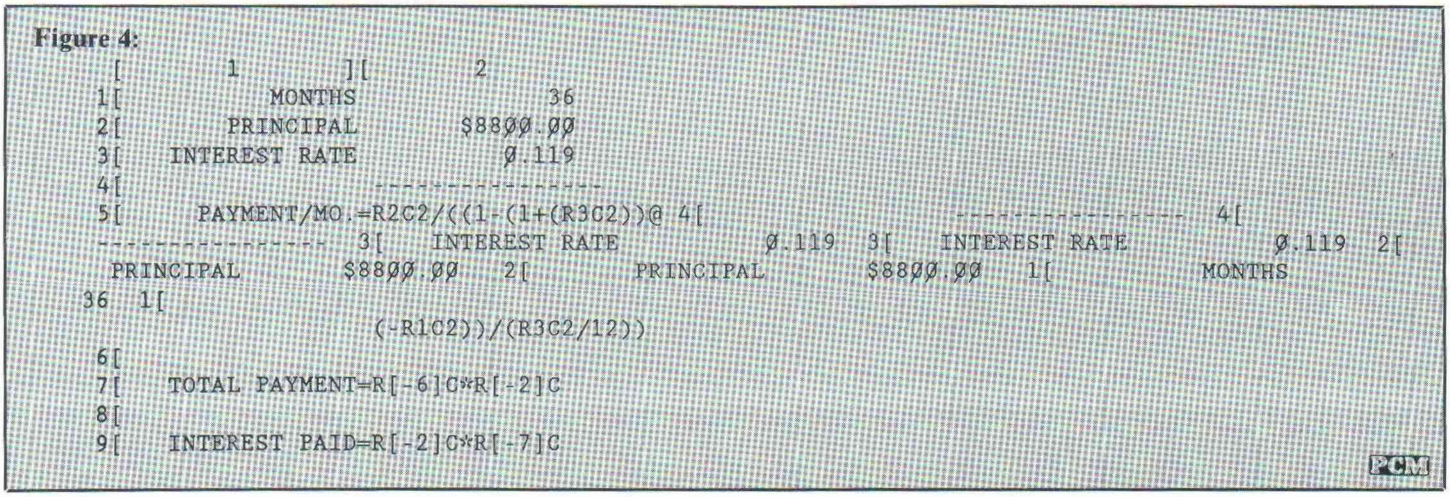


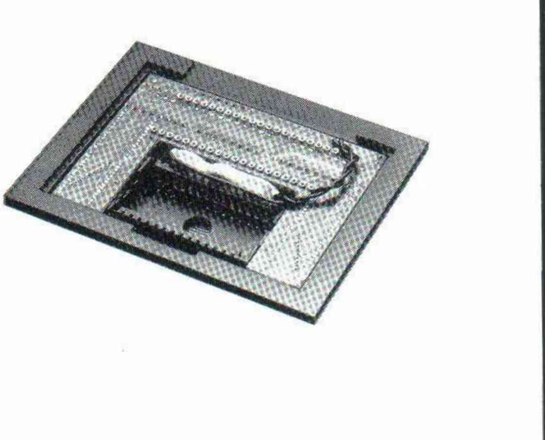
Figure 4:

```
     [        1       ][       2
    1[          MONTHS              36
    2[       PRINCIPAL        $88ØØ.ØØ
    3[   INTEREST RATE           Ø.119
    4[                 ---------------
    5[     PAYMENT/MO.=R2C2/((1-(1+(R3C2))@ 4[                  ---------------  4[
    ---------------  3[   INTEREST RATE            Ø.119  3[   INTEREST RATE            Ø.119  2[
     PRINCIPAL       $88ØØ.ØØ   2[       PRINCIPAL        $88ØØ.ØØ   1[          MONTHS
   36  1[
                         (-R1C2))/(R3C2/12))
    6[
    7[   TOTAL PAYMENT=R[-6]C*R[-2]C
    8[
    9[   INTEREST PAID=R[-2]C*R[-7]C
```

Give your Model 100

Installs as easily as plugging in a socket

96K RAM



PCSG says: Satisfaction guaranteed or your money back within 30 days

As amazing as it seems you can upgrade your Model 100 to 96K of RAM in just 60 seconds.

It comes to you right out of the box looking just like the picture. You just open the little compartment on the back of your Model 100 with a quarter and it just pushes right into place. You can then put the cover back in its place.

You then have 3 banks of RAM of 32K each. The additional two banks also work just like your Main Menu.

You just push a function key and you look at the second bank. Push it again and you are in bank three. Press it one more time and you are back to your original bank.

It has its own built-in NiCad battery that recharges right from the Model 100 and its guaranteed for a full year.

What is really great is that you can copy a file from one bank to another with just a function key.

Each bank is like having another Model 100, and all the built-in programs as well as any snap-in ROM programs appear in all three banks and work the same way. Your widebar cursor moves from file to file and you access any file or run any program just by pressing ENTER.

What lets you copy any file from one bank to another is a snap-in ROM from PCSG called RAM+, that comes at no extra charge. It just pushes right into the little socket in that same compartment with the 64K expansion unit.

Not only does this firmware let you copy a file from bank to bank, but you can make a copy of any file within the same bank instantly with a function key. Great for Lucid spreadsheets!

Copy a file from bank to bank with a function key

You can also rename a file, or kill any file with just a function key. Plus you can do a whole lot of other useful things like setting the date, day and time with function key ease. You even have a function key that lets you use non-Radio Shack printers without having to make those tricky dipswitch settings.

RAM+ lets you cold start any one of your banks without affecting the other two. That means that anytime you want you can clean out a bank's entire memory, but leave intact all the files in the other banks.

What is also fantastic is that you don't have to have the ROM in place to use the additional RAM. Whenever you take out the snap-in ROM it leaves behind a tiny machine code program that lets you switch from bank to bank just by pressing ENTER.

This lets you use your ROM socket to snap-in other ROMS like LUCID spreadsheet, WRITE ROM text processor, or DISK+ ROM file transfer program, and use them in any or all three banks. All of these, by the way, are available from PCSG.

When you are ready to copy a file from one bank to another or use any of the other fantastic functions we talked about you can just snap the RAM+ ROM back into place.

Everybody that has this 96K system in their Model 100 is so excited, because it gives them three times the capacity and all banks work just like the Main Menu.

And what has made a lot of people happy is that the system bus, located in the same compartment, is left free for you to plug in a DVI or the Holmes Engineering/PCSG portable disk drive.

The ability to copy a file from bank to bank instantly with a function key, plus all of the other features make this RAM extension truly an engineering masterpiece.

Some people hesitate when they think of installing something, and then others are skeptical that any additional hardware could be as good as the Model 100 itself. That's why we sell these 64K expansions on a 30 day trial. Simply return it within 30 days for a full refund if you are not satisfied. Priced at $425. MC VISA COD.

1-214-351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. No. 207, Dallas, Tx. 75229

VISA / MASTER CARD / C.O.D.

HARDWARE

The Plot Quickens With HI's Digital Plotter

Upon first glance, the Houston Instrument Model 695 plotter looks like a toy. But upon working with the plotter, you discover that it's no toy. Its light weight and unique plotting mechanics make this plotter more efficient, faster and less expensive than other plotters in its class. Additionally, the Model 695 accepts Houston Instrument's standard "Digital Microprocessor Plotting Language" to make it software compatible with many other existing plotters.

The Model 695 plots on the two most popular paper sizes, 8.5 x 11 (letter) and 11 x 17 (legal). The international model accepts sizes A and B paper. You can use any high-quality bond paper or transparent film. Special pens are required for transparencies.

Eight hard-nib pens are provided with the plotter which can automatically switch between four colors. Pen wells are conveniently built in for storage of the remaining four pens.

The Model 695 PC Series Digital Plotter connects to your computer via the RS-232 (serial) interface. Model 1000 and 1200 owners will need to add an RS-232 board to their computer before connecting the plotter.

At 0.001 inch, 0.005 inch, 0.1 mm, or 0.025 mm, the addressable registration on this plotter is similar to most plotters of this type. What was so amazing about this plotter was its speed. Using a combination of paper and pen movement, the pens can move across 3 inches per second axial or 4.2 inches per second diagonal.

When you insert paper into the plotter, it seems to go haywire as it rolls the paper back and forth. What it's doing, actually, is sizing the paper. There is no need to tell it what size you're using — it automatically computes the size and a corresponding light on the indicator panel illuminates.

The manual provided with the Model 695 plotter provides clear instructions for setting up the plotter and attaching it to your computer. The more technical aspects dealing with writing your own software for the plotter are in a separate section of the manual. They even have a semi-technical trouble shooting guide for correcting common hardware problems. A parts list is included.

The programming section of the manual has done very well what most other hardware manuals do poorly if at all. They have managed to include tutorial instruction without getting in the way of the facts — all while keeping it to a short, easy-to-read section of the manual. Most manuals either lean too much to the tutorial side, making it virtually impossible to use as a reference guide, while others lean to the technical side, providing no helpful hints or demonstrations. I would have liked to see a few more examples, but, all in all, the manual is neatly organized and a pleasure to use.

Most people will use this plotter for applications such as Auto-CAD or *Lotus 1-2-3*. The IBM (Tandy 1000/1200) versions of these pieces of software support this plotter. Since it supports the DM/PL (Digital Microprocessor Plotting Language) standard, it will be compatible with many other software packages.

Since Tandy 2000 software was written specifically for the Tandy 2000 and other Tandy hardware, the Tandy version of *Lotus 1-2-3* available at press time did not support this plotter.

What you must do before you go out and buy one of these beauties is to check the software you plan to use first. If it supports a Houston Instrument plotter with DM/PL, chances are it will work with the Model 695. If it doesn't, you must either write your own driver for the plotter (usually difficult) or use it exclusively with software you write yourself.

If you decide to write your own software for the Model 695, you'll find a full compliment of plotter commands available to you. In addition to common commands to move the pen, draw lines and change colors, more advanced commands take advantage of the Model 695's internal processor to draw arcs, draw ellipse plots, draw text and set up logical windows and viewports on the plotting surface.

If you are in the market for a high-precision graphics output device, the Houston Instrument PC Series Digital Plotter Model 695 is a good choice. At $799, its ease of use and performance make it a bargain.

(Houston Instruments Inc., 8500 Cameron Road, Austin, TX 78753, $799)

— Danny Humphress

SOFTWARE

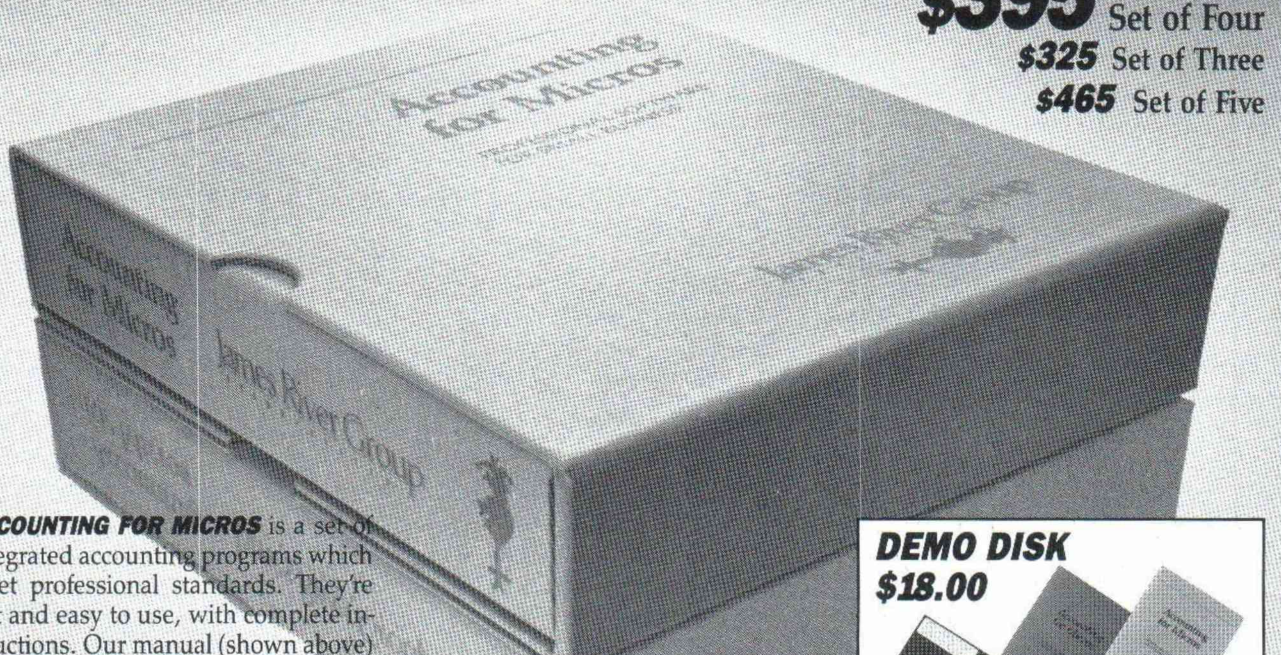
Account Mate II Has Innovative Differences

By M.J. Batham

The *Account Mate II* software consists of six individual modules: *General Ledger, Sales Order, Purchase Order, Accounts Payable, Invoice/Accounts Receivable* and *Payroll*. Data from each component may be integrated into the *General Ledger*.

ACCOUNTING FOR MICROS

$395 Set of Four
$325 Set of Three
$465 Set of Five



ACCOUNTING FOR MICROS is a set of integrated accounting programs which meet professional standards. They're fast and easy to use, with complete instructions. Our manual (shown above) also includes helpful information on bookkeeping and computers.

GENERAL LEDGER $125
Allows up to 1,000 accounts & 1,000 transactions/month. Retains mo/end balances for Last year, This Year and Forecast. Includes Cash Disbursements, Cash Receipts and General Journals. Reports include Balance Sheet, Income Statement, Annual Summaries and Journal Reports.

ACCOUNTS RECEIVABLE $125
Allows up to 2,500 customers and 1,000 invoices per month. Invoicing can access Inventory Module. Keeps customer names and addresses. Invoice prints on plain paper or any pre-printed form. Statements can be printed at any time.

INVENTORY $125
Allows up to 4,000 parts. Keeps 3 month history of unit sales as well as year to date. With AR, can be used as point of sale system (prints invoices, handles cash). Reports include Inventory Value and Stock Report, Internal and Customer Price List.

ACCOUNTS PAYABLE $125
Allows up to 500 vendors and 600 invoices/mo. Records invoices and handwritten checks. Prints computer checks on any pre-printed form. Keeps vendor names and addresses.

PAYROLL $125
Will handle up to 100 employees with six deductions per employee. Deductions may be determined as fixed dollar amounts or percentages, or referred to a table for automatic look-up. Tax tables are easily entered, or purchased separately.

SET OF FIVE $465
SET OF FOUR $395
SET OF THREE $325

RUN ON MOST CPM AND MSDOS

Apple CPM
Columbia
Compaq
Corona
Eagle (all)
Epson QX-10
IBM PC,XT,PC jr,AT
Kaypro (all)
Morrow (all)
Osborne (all)
Panasonic
Radio Shack CPM
Sanyo (all)
Tandy (all)
TeleVideo
Zenith 100 & 150
8" CPM
Other compatibles

DEMO DISK
$18.00

Try all 5 programs above (GL, AR, AP, IN, PR). Order our DEMO DISK for $18.00 (includes shipping). Condensed versions of the programs give you the "feel" of data entry and access. Includes sample reports and instructions. Specify machine.

TRASHMAN (NEW) $145
The "Catch-All" program. Files any type of information for quick access. Name or subject oriented with 15 lines of notes per name. Use TMAN as a mailing list, filing system, notebook, etc. Can be used alone or with data from our other programs.
Try **TRASHMAN (only) DEMO $16**

HOW TO ORDER: Please specify machine and disk format. You can pay by check, by VISA or MasterCard (we need your expiration date and card number), or by UPS COD (add $2.50 COD charge). Our price includes shipping. Minnesota residents, add 6% sales tax). We ship most orders the same day.
or **ORDER BY PHONE: 612-339-2521**

James River Group
INCORPORATED

(612) 339-2521

125 North First Street
Minneapolis, MN 55401

Hardware requirements are a Tandy 1000, 1200 or 2000. *AccountMate II* runs in monochrome on a color monitor, and is not copy protected. A hard disk is necessary to merge the programs, but a dual drive floppy disk system is adequate if only one package is used. Three formats are supplied: a compiled version in machine code which requires 192K RAM, a *dBASE II* Version 2.4 or higher which runs on 128K computers, or a *dBASE III* Version which needs 256K.

All reports print out on an 80-column printer. Most accounting programs require a larger carriage or print in condensed mode on a smaller printer. The acceptance of an 80-column printer is very useful.

AccountMate II is different from the myriad of accounting software clones out there. I tested the compiled version which resembles *dBASE* in screen format. The menus were so simple to use I needed the user's manual only once to start the program. The main menu gives four choices: posting, maintenance, reports and miscellaneous.

AccountMate II has a unique method of installing the general ledger system. Rather than cluttering up the main menu with a separate start-up option, the opening screen shows the current date. To initialize the program, type `99/99/99` for date and the software proceeds to the setup procedure.

The $395 *General Ledger* module handles up to 65,000 accounts and has many innovative ideas, including the ability to open a new account while in the posting menu. Many times when posting a customer's books, I discover a new account has been added. I must post the entry to a miscellaneous account, wait for the transactions to print out, go back to the main menu, add the account, and return to the posting function to credit the miscellaneous account and debit the new account. *AccountMate II's* new account addition while posting really saves time.

A nifty feature is instant recall of the chart of accounts while in the posting menu. Entering a question mark displays the accounts in numerical order. A range of accounts is specified by typing `75?` and all accounts greater than 75,000 are displayed.

Data entries can be posted for this month or for a previous month. After an account number is typed, the screen displays the account description and month/year-to-date balances. Check number, a short description of the transactions and date (defaulting to today) are entered.

On-screen editing is done immediately. Zeros to the right of the decimal point are not necessary, so a hundred dollars is posted quickly as 100. Credits are indicated with a minus sign. The program accepts entries with commas such as 10,000 which is really helpful when posting.

The multiple account posting feature is excellent. A source account, such as cash, is credited and up to 99 different expense accounts are debited. Transferring from a manual to an automated accounting system in midyear is simplified with this procedure.

After transactions have been added, an audit menu allows an entry review with scroll backward, forward or jump scrolling. Incorrect postings are deleted and any or all of the transactions may be printed to provide an audit trail.

The maintenance options show account balance totals and account status. Budgets can be set up for each account. Amounts are posted automatically for semi-monthly, monthly or for specified periods, or distributed on a percentage basis to two or more accounts.

Reports may be displayed on the screen or printed out. Most other programs don't offer the screen display option. A dual password protection scheme allows the person posting to access part of the data, but a second password is required for the manager to view the financial statements.

The miscellaneous section of the menu allows end of month processing, a review of data file updates or a change in system setup. A really great feature is duplication of the chart of accounts for different departments or companies. Rather than typing every account twice with a separate extension, the chart is replicated which saves hours of time.

I was getting error messages when starting the program. The package was installed in a sub-directory on my hard disk, and I hadn't copied the special SofTech Information Systems "config.sys" file in my boot directory. My MS-DOS config file defaulted to 10 open files, but *AccountMate II* searches for 16 files. The SofTech Information Systems technical support staff were very helpful and the programmer answered a few of my more technical questions.

Although the *AccountMate II* General Ledger program uses a lot of memory for the system files, it performs very quickly and efficiently. This is a very powerful program. I was impressed with the ease of setup, and the on-screen help file.

What makes *AccountMate II* different from other programs is the choice of displaying financial reports on screen as well as on the printer, easy to use menu formats, a resident help file and a beginning chart of accounts with major account categories already numbered.

The ability to add accounts while in the posting menu enabled me to set up a company's books in less than a half hour — the fastest installation time yet for a general ledger program! A non-accountant manager will really appreciate the simplicity of using *AccountMate II* — even those of us jaded accountants thank the authors for the ease of operation.

When ordering *AccountMate II*, advise SofTech Information Systems which machine you will use, rather than ordering the MS-DOS version which caused a strange screen display. My Tandy computers emulate *AccountMate II's* PC-DOS version and that program worked fine.

The *Sales* and *Purchase Order*, and *Accounts Payable* and *Receivable* modules will be reviewed in a later issue. A demo version of *AccountMate II* is available from the software authors for $25.

(Softech Information Systems, 20 Sunnyside Avenue, Mill Valley, CA 94941, 415-381-1011, *AccountMate II General Ledger*, $395, *Accounts Payable* or *Receivable*, $295 each, *Sales* or *Purchase Order*, $195 each)

SOFTWARE

pfs:file/report Ease and Flexibility Within Limits

By Victor Scheluchin

Although sold separately, *pfs:file* and *pfs:report* together make up a versatile file management system designed to be easy to learn and use. This system is intended for people who don't want to become involved in the complicated process and technical aspects of more powerful (and correspondingly more expensive) database management programs. Judging by the long-lived popularity of the *pfs* series, which is distributed by Tandy, Software Publishing Corp. has been extremely successful in achieving its goals.

These programs' ease of use, however, must be weighed against a few limitations. *pfs* works with only one file at a time and is organized similar to a card file or a file of paper forms. This means you cannot perform searches or generate reports based on the contents of more than one file at a time.

Also, *pfs* stores information on disk in a unique way. If you already have files created by another database program, you won't easily be able to transfer them over to *pfs*. The same is true if you ever need to use files created by *pfs* in any non-*pfs* program.

Each program has a main menu and several submenus. The choices are clearly labeled, and are selected by entering their numbers. *pfs* uses the F10 key to execute a command instead of ENTER, which most other programs use. If you press ENTER out of habit, the cursor simply drops down a line and moves to the left side of the screen. This is true for data entry screens as well as menu screens.

You can cancel what you're doing and return to the main menu at any time by pressing the ESC key. The programs use the TAB key to move from one item on screen to another. I found this to be an unfortunate design flaw in an otherwise well-crafted system. TAB and ESC are right next to each other on the keyboard, and it is easy to lose some work by accidentally pressing ESC.

A help screen can be accessed from inside most of the menu choices by pressing F1. The help screens outline the various options available and, because these programs are so easy to use, contain almost enough help to make the manuals unnecessary after the first reading.

The manuals are attractively typeset and do an excellent job of explaining the setup and operation of both programs. Both manuals are extensively illustrated with actual screen contents and paper form and report layouts. Each also contains an index, glossary and appendices. Two things missing, though, are any mention of how to get rid of unwanted attachment pages, and some suggestions for using the *pfs* programs and files with non-*pfs* programs.

As usual with Tandy-supported software, a software registration card must be filled out if you want to be notified of changes or updates to the programs. In addition, Software Publishing Corp. includes enrollment cards to their User Groups.

Both *pfs:file* and *pfs:report* are copy-protected, but the company provides a spare copy of each program disk. Unfortunately, a hard disk user must load the programs from diskette every time they are needed. Once a program is loaded, however, the program disk can be replaced by a data disk and files can, of course, be stored on both floppy and hard disks.

pfs:file

pfs:file is a basic, easy-to-use electronic filing system, organized as similarly as possible to the traditional card or paper file. Most database programs talk about "fields" and "records." The *pfs* manuals refer to fields as "items of information" and to records as "forms."

Designing and setting up the blank electronic form on your screen is very easy and well-illustrated by the examples in the manual. You simply type the names of the items you are going to store wherever you want them to appear on the screen. The few possible pitfalls here are clearly explained and therefore easy to avoid.

If you like, you can set up your forms on screen to match the layout and appearance of paper forms. For really long forms, *pfs:file* allows you to have up to 32 pages of screen forms. In addition, attachment pages can be added to any form at any time, to hold any extra information which doesn't fit the normal layout. Once your form is set up, you can begin entering information into your database.

Though the manual claims up to 5,000 forms can fit on a Tandy 2000 diskette, a more realistic figure is given as 2,500 names and addresses in a typical mailing list. The manual includes a very good explanation of how to determine the maximum size per file. A form can have up to 32 pages of up to 100 items per 21-line page. The larger the form, of course, the fewer forms you can fit on one disk. The larger the file, also, the longer it takes to complete a search.

To search for a record, you select the Search/Update function from the main menu and then fill in a blank form according to the item or combination of items you are searching for. You can search for partial as well as exact matches as well as numeric ranges. When a record matching your search conditions is found, *pfs:file* displays the form on screen and you can then make changes to the information or continue the search.

pfs:file is a good, flexible, elementary database program which allows you to store and then retrieve information on screen. Its ability to print reports, however, is rather limited. You can print or omit specific items from each form and specify whether you want to begin a new line after an item, but you cannot rearrange the order of the items, or sort by more than one item without redesigning your entire form. You can store up to eight simple print formats per file.

pfs:report

For anything more than mailing labels or simple printed listings of your database, it is necessary to use *pfs:report*. With *pfs:report*, you can design up to eight report formats for each file. These formats can be saved and used later to print your information sorted numerically and alphabetically. You can get column totals and subtotals, counts of column entries, and neatly aligned decimal points. *pfs:report* can also do calculations on numeric items.

pfs:report can print from one to 20 columns in any order, sorted by whichever item is placed in the first column. If the first column contains duplicates, *pfs:report* will then sort by the next column. You can choose to include only certain items from each form, or only forms meeting specific conditions. The conditions are specified exactly as in *pfs:file*'s Search/Update function. Reports can be printed to either parallel or serial printers, the display screen, or to a disk file. Like *pfs:file*, *pfs:report* will not work with non-*pfs* files.

These advantages must be weighed against two major disadvantages: first, the possibility of not having access to your files for a while if anything happens to your two copies of the programs. And second, the inability to transfer information into or out of the files used by these two *pfs* programs means that once you decide to go with *pfs*, it's *pfs* from now till forever, or at least until you can find a program which will accept the unique *pfs* file format.

At $265 for the *pfs:file/pfs:report* combination, the programs are, by MS-DOS standards, moderately priced. For the typical user who has moderate needs, they are a good buy.

(Tandy, 1700 One Tandy Center, Ft. Worth, TX 76102; *pfs:file*, Cat. No. 26-5305, $140, *pfs:report*, Cat. No. 26-5306, $125.)

SOFTWARE

My Word! Almost A Free Lunch!

Both with my Model I and now with my Tandy 2000 I have looked for the cheapest software available to do the job. *MY WORD* at $25 (or $35 for the extended version) certainly is my kind of program. The next cheapest word processing program for my Tandy 2000 I have ever seen was about $200.

But how good can a $25 word processor be? I think it is very good and it is being used to prepare this review. The manual is full size, 8 by 11 inches, and runs about 70 pages. Except for one oversight which I will describe later, the manual is quite good. Bruce Tonkin, the author, even makes allowance for those who don't like to read the manual. The first short chapter tells you just enough to get up and running. Following that, the manual goes into detail of all the features.

MY WORD is not copy protected and the manual gives very good instructions on how and which files to copy to your work disks. In an effort to leave as much room as possible for your text, *MY WORD* comes in two basic pieces. One file loads as the actual text editor and the other file is called to actually print the finished document. While this is a minor inconvenience, the editor program does contain a quick print feature that will give you a hard copy without fully formatting the text.

The manual contains some comments by the author about how the program was written, why some of the features are as they are and admits freely that this program is modeled closely after another well known word processor, "WS." Obviously, he means *WordStar* and there is space devoted to the similarities and differences between *MY WORD* and *WordStar*.

MY WORD presents you with a totally blank screen when you start a new document. You type and the letters appear on the screen. The power of a word processor is massaging the shape of your text and *MY WORD* is told how to do that by dot commands embedded in the text by the user. This approach means that you must learn those dot commands, but they are easy and mnemonic.

Any line that begins with a dot (period) in the very leftmost position is taken by *MY WORD* to be a print format command line and the text on that line will appear on the screen, but will not be printed. Unknown commands are ignored so this feature can be used to insert comments in the text that will appear on the screen but not on the printout.

For example, if you wanted to have your final printout double-spaced, typing `.sp2` all the way against the left margin would cause all lines following the spacing dot command to be double spaced. All the margins of your document, right, left, top and bottom, are controlled by dot commands. In addition, headers, footers, page numbering and merge printing commands are entered by dot commands.

While some study of these commands is required to use *MY WORD*, there are only 27 and the names make sense.

I have one complaint and one warning for the user about the dot commands. Only one command can be put on a line. That means that it may take the first seven or eight lines of the document to set up the format. Maybe I am picky, but it would be very nice to be able to put multiple commands on one line.

The warning concerns a problem that gave me fits. The `.rm` command determines the length of the text line. If you set `.rm 60`, then any text will wrap to the next line after 60 characters have been entered on a line. My problem came when I changed the `.rm` after entering several lines of a document. After changing the margin, I used the CTRL-B command to reformat a paragraph. Nothing happened. I went to the print program and printed the text. There was still no change in the margins.

In frustration, I saved the document and went to bed. The next day after the document had been read back into memory, the paragraph reformat function worked. I changed the margins again and reformatted. There was again no response.

One very nice feature of *MY WORD* turned out to be the solution to my problem. Pressing the ESCAPE key during editing displays useful information on the bottom line of the screen. Based on the cursor position, the current page number, line on that page, line in the document, total lines in the document, column the cursor is in and the current right margin are reported. This information is not updated with each keystroke. The author contends that the slowdown is not worth the benefit. Pressing the ESCAPE key will instantly update the information.

It finally dawned on me that ESCAPE must go look for the last `.rm` command. Sure enough, changing the margin, pressing ESCAPE and then doing a reformat works every time. I have not found any reference to this problem or the cure in the manual.

While on the topic of one-key help, *MY WORD* keeps one screen full of help in memory. Pressing F1 clears the text and displays the name of the current file, the time, memory available, memory used by the document and the insert/overwrite mode in effect. In addition, the use of keys F1 through F10 is described. Saving your file, saving and exiting, renaming and several other useful tasks are available with one function keystroke. At the bottom of the screen, 20 common editing commands are listed as reminders. F2 clears the help and puts your document back on the screen.

I don't want to describe everything *MY WORD* will do and you don't want to read that list. Suffice it to say that *MY WORD* will do all of the normal word processor things you expect from the $200 models. Merging of data from a file into letters and sorting within a block of text are extras included in *MY WORD*.

I was also delighted to find that the usual handling of blocks of text — copying, writing to disk, deleting — were supported. Although the maximum size of a single document is about 48,000 characters, the link file, `.lf` command, allows multiple files to be chained together as one document. The page numbering and formatting carries over from one file to the next so it looks like one file when printed.

The bottom line is that *MY WORD* is a very capable program at any price and most users could go a long time without finding something it won't do. At $25 or $35, *MY WORD* approaches "free lunch" status.

(TNT Software, 34069 Hainesville Road, Round Lake, IL 60073, $25, extended version $35)

— Potter Orr

SOFTWARE

Monopoly is a Fun Ride on the Reading

Ever had the urge to play Monopoly and no one was around to play with, or you couldn't talk anyone into playing? Well, the answer to your problem is here! *Let's Play Monopoly* by Custom Software is a fun and challenging game that pits you against your Model 100 (24K is required).

The game is the same as the popular board game (it is assumed that you know the rules of the game because none are included). The only exception to the rules that I could find is that you don't have the opportunity to purchase property that the computer lands on but doesn't purchase.

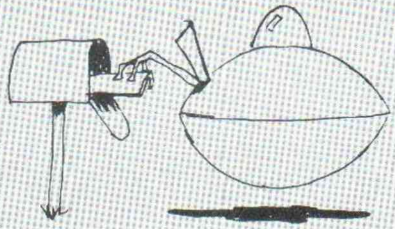
At the beginning of each turn, your options are displayed: ROLL the dice, BUY houses, MORTgage or UNMORTgage property, TRADE with the computer, REVIEW or QUIT.

On the screen, the monopoly board is displayed with each property looking the same. The provided instruction card shows the same board with each square labeled by number or type. For example, Oriental Avenue is labeled a '2' because it is in the second group of properties, 'R' is for railroad, etc. The only problem with this is that you can't tell exactly where the property is without consulting a Monopoly board. (This is especially frustrating when dealing with the railroads, since the card doesn't specify which railroads are which). However, as you gain familiarity with the game this becomes less of a problem.

Let's Play Monopoly has its assets, also. It uses graphics to draw a picture of each card when you land on that square, including the "Go to Jail" square. It's challenging! It has all of the same options as the board game. You can buy houses (five make a hotel, one hotel limit). The computer can build houses — and it can lose them, too. You land on Chance and Community Chest and get those "Go to Jail" and the "Advance to Go" cards. You can mortgage property (better watch what you mortgage; the computer will let you mortage everything — including the kitchen sink — whether you need the money or not). You can even trade properties with the computer, but the computer only trades for a monopoly (and who doesn't?).

There are some features which I wish the program had; such as alerting you at each of your turns how many pieces of property you have mortaged, if any. As it is, you may end up playing half the game with big bucks without realizing that you have a property mortaged until the computer lands on that property and you lose the rent.

The game is equipped with a review option to look at the status of all of



the property, including your property, the computer's property and any unowned property. But, the hitch is to remember to look! The one thing I wish were included in the review section is the current rent on each of the properties displayed.

This game is good to occupy you during those times when you're waiting for the professor to get to class, or when you're riding the commuter train or just sitting around the house without anything to do. In my case, it occupied me during Christmas break (it was a life saver!). The game is equipped with a save feature which allows you to come back to the same game or start a new one.

The game is very fast moving so you have to pay attention to what is going on. There are several reasons why the game goes so fast. One reason is that the computer takes care of paying rent, adding and subtracting the appropriate amount, and calculating the amount you own when it's tax time. It takes around an hour to an hour and a half to play a good, "hard hitting" game. Although the instructions say you will be doing good to win half the time, I won three of the first five games I played.

The documentation is enough to get the program loaded (although it doesn't say the game requires 24K). It explains what the board looks like and what the choices on the player's menu mean. However, it doesn't go into detail concerning the rules of Monopoly; as already mentioned, they assume that you know them by heart.

In conclusion, *Let's Play Monopoly* is fun, entertaining, and a challenge. It's nearly everything that the board game is — only it is portable and fast.

(Custom Software, 1308 Western, Wellington, KS 67152, $29.95)

— Kerri L. Arnold

SOFTWARE

Dr. Preble's Flying Machine 2.2

About this time last year, I wrote a review of Dr. Preble's *NavComp* software (PCM, May 1984). This spring I was given a chance to review the second version of Dr. Preble's navagational-flying program. I will in this review cover only those items that are new or substantially changed, referring you to the previous review for more complete information.

There are still two programs supplied on cassette and both require a little over 8K of memory plus room for two text files. The programs, *NavPln* and *NavAid*, can be stored together or separately as before. The first program, *NavPln*, allows you to plan the flight, usually at home or on the ground where sectional or IFR charts are spreadable. Entering information is really a snap this time since the menus have been clarified. One of the few drawbacks to the earlier program has been solved: the ability to back step through the menus to correct erroneous entries (would we really make them if we were flying?). One of the best improvements was in the area of the flight plan storage requirements. It appears that this version has very significantly reduced and compressed the flight information. It also took a lot less time to save the flight plans to the two .DO files.

Once the information is entered, it's time for the real flying world, probably my favorite piece of software anywhere. The *NavAid* program is the real time software that interprets the data you've entered. The main menu has been revised and is very clear and quick. One of the hassles of the old program was that when you went to one of the windows for information (for instance "how much time is left?") the clock did not change. Now all ETA and ETE are updated as fast as the screen can print them. Several times I tried to trick the program, like flying too fast or gaining altitude too fast, and I got percent signs by the speed. This version also seems to trap errors in data entry better. We flew from Louisville to Nashville in ground school using this version and found it would handle almost any changes in speed, directions, winds, and VORs efficiently and quickly.

Last but not least is an improvement in the instruction manual. Each item for data entry is very well described, thereby significantly reducing errors.

If you fly and have not already purchased Version 1, then take a good look at this updated version. It even comes with a 30 day full refund.

(Doctor Preble's Programs, 6540 Outer Loop, Louisville, KY 40228, $34.99 plus $1.50 S/H)

— Vincent Lord

Portable Disk Drive

Portable Computer Support Group in association with Holmes Engineering announces the "Chipmunk" portable disk drive, to expand the capabilities of the Model 100 and its cousin, the 200. In keeping with the 100's "main menu" concept and function key operation, the disk drive appears to the end-user like an extension of RAM.

Owners used to being limited to 32K of RAM can find they have 358K of storage space on any given 3½-inch diskette. The Portable Disk Drive weighs 3½ pounds, lets users store over half a megabyte on two diskettes and comes with six programs allowing users to use the disk drive and portable computer in new ways, such as "live" downloading from *Telcom* to disk. The "Chipmunk" retails for $599. For more information, contact Mike Anders, P.C.S.G., Inc., 11035 Harry Hines Blvd, No. 207, Dallas, TX 75229,301-248-9090.

Tandy 1000 Multifunction

Hard Drive Specialist announces the TanPak multifunction board, designed specifically for the Tandy 1000. It contains special DMA circuitry, Memory, Serial Port and Clock.

The TanPak is offered in 128K, 256K, or 512K versions, with the smaller versions being upgradeable with 256K upgrade kits, and software to support its features. The 128K version costs $399. For additional information, contact Hard Drive Specialist, 16208 Hickory Knoll, Houston, TX 77059. 1-800-231-6671 or 713-480-6000.

A New Telecommuter

Sigea Systems, Inc., introduces *Telecommuter—plus*, a new product for the word processing and communications user. It has all the capability of *Telecommuter* with these new features: 1) The word processor allows access to DOS from the edit screen. Users can sort their ADRS.DO file after making new entries, rename a file, create or remove a new subdirectory, or perform almost any other DOS operation while using *Telecommuter's* word processor; 2) *Telecommuter—plus* has XMODEM to permit the exchange of text and binary files using protocol checking to eliminate errors from noise on telephone lines; 3) Automatic redial of data numbers such as bulletin boards. One keystroke starts periodic dialing until modem detects a carrier or the user terminates manually; and 4) Host mode has eight different access levels. Users can restrict the remote caller to the current subdirectory, and prevent the caller from writing over or erasing existing files or using MS-DOS commands.

Both *Telecommuter* and *Telecommuter—plus* work with the new Radio Shack modems, including the DC-12212 and the Tandy 1000 board modem. Users choose the proper modem type on *Telecommuter—plus'* installation screen and there are no other setup adjustments. *Telecommuter—plus* is available for $400. For more information contact Sigea Systems, Inc., 19 Pelham Road, Weston, MA 02193, 617-647-1098.

One Step Telecommunicating

Switch & Mux, Inc. has released a software package for the Model 100 and 200 called *One Step Communications.*

One Step Communications is designed for newspaper reporters or writers working in the field. All the reporter has to do to transmit a text file is enter the program and type in the name. The program automatically sets up the *Telcom* for acoustic operation. It also sends a special ANPA high-speed wire header before the text file and ETX EOT after the text file. The program allows the user to send more than one file in a single transmission to a host system. When users are finished sending files they simply exit the program and it terminates the communication. *One Step Communications* retails for $100. For more information, write: Switch & Mux, Inc., 10 Oakridge Avenue, Merrimack, NH 03054, 603-424-4161.

File Management System

Executive Systems, Inc., announces *XTREE*, a new MS-DOS and PC-DOS file management program which facilitates the file and directory maintenance process. *XTREE* provides commands to access, delete, rename, view, move, list, or show any and all files within any and all directories on a floppy or hard disk.

XTREE's initial screen is organized into nine separate sections. The first six display directories, subdirectories, files and valuable statistical information. The final three provide available commands, control and function keys. At any time, users can see every file on a disk, regardless of the directory it is in.

XTREE requires an MS-DOS 2.0 or PC-DOS. The maximum number of files a user can control is 2800; the maximum number of directories is 180. The program retails for $49.95.

For additional information contact: Executive Systems, Inc., 15300 Ventura Blvd., Suite 305, Sherman Oaks, CA 91403 or call 818-990-3457.

Floppy Diskette Storage

Potomac Industries, Ltd., offers a new 5¼-inch floppy diskette holder, DiskPorter. This slim-profile storage case is 1½ inches deep and just under a foot square. It offers a stairstep design to keep the labels exposed on the 20 diskettes it stores and direct access, eliminating the need to constantly flip through files. With its cover flipped back as an easel stand, it doubles as a copyholder without wasting work space; closed, it can lock into a desk drawer for maximum security. Index cards slip behind its clear slope-front cover to make it easy to select the right diskette from a case, the right case from a stack. DiskPorter diskette organizers are available for $29.95. For more information contact Robert Rager, Potomac Industries, Ltd., 2300 M Street NW, Suite 400, Washington, DC 20037, 202-955-9797.

MEGABYTES FOR MINI-BUCKS

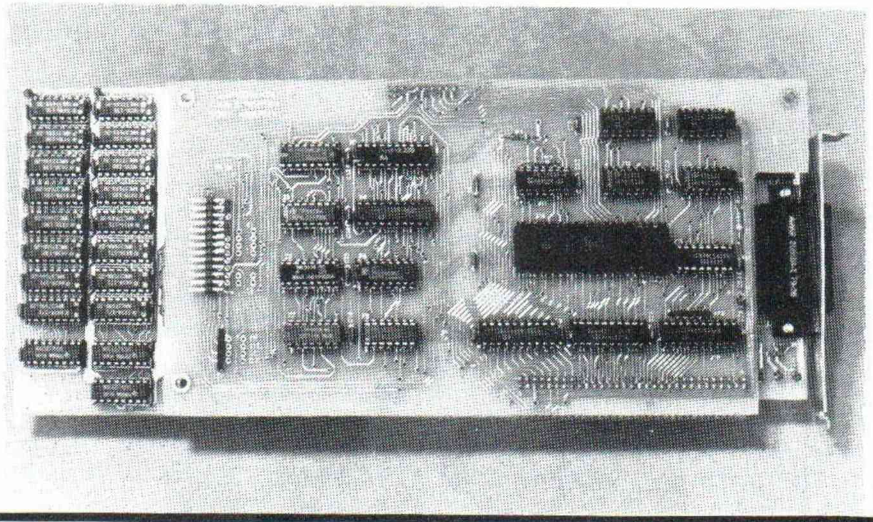
Tandy 1000 Hardware

Tandy 1000

TanPak

The first and only board that your Tandy 1000 may ever need. Your 1000 is very versatile, new applications and functions are being developed for it every day. And now with the HDS TanPak you can keep your options open for tomorrows technology. As you know the Tandy 1000 only has 3 expansion slots, and those are not quite PC compatible. The TanPak does the job using one expansion slot that normally would require four using Tandy boards. The TanPak includes a Serial Port, a Clock, and sockets for up to 512K of Memory Expansion. And the best part of all, it will save you some big bucks over the Tandy boards.

TanPak 128K . . . . $399.



Tandy 1000

TanTel

300/1200 Baud Internal Modem

Both 300 and 1200 Baud within the 10" that is alloted by Tandy for a board. But that's not all. This internal modem also features automatic dialing, an auto-answer feature, full or half duplex, automatic speed selection, and a speaker to alert you to busy signals and wrong numbers. FCC registared and full documentation included.

Internal Tandy 1000 Modem "TanTel" . . . $299.

AS SEEN AT THE RADIO SHACK COMPUTER SHOWCASES

Disk Drives

Fast stepping add on or replacement disk drives for half the Radio Shack price . . . . **$149.**

Tandy 1000

Eprom Programmers

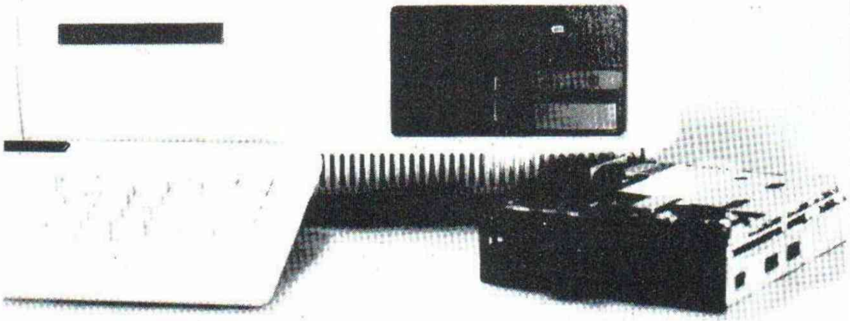
These EPROM Programmers plug into the Model 1000 edge card slot and complete with software and documentation. The 24 pin programmer will program the following EPROMS: 2704, 2708, 2716, 2732, 2508, 2516, 2532, 68764, and 68766. The 28 pin programmer will program the following EPROMS: 2764, 27128. and 27256.

24 Pin Programmer	**$149.**
28 Pin Programmer	**$149.**

Hard Drive Kit

Put a 10 Megabyte Hard Drive inside your Tandy 1000. This Hard drive subsystem includes an interface card and a Half-Height Hard Drive that directly replaces a floppy disk drive in both size and power consumption drive.

10 Megabyte Internal Drive System	**$895.**
10 Megabyte External Drive System	**$1095.**
15 Megabyte External Drive System	**$1395.**
30 Megabyte External Drive System	**$1895.**



Radio Shack and Tandy are trademarks of Tandy Corporation, TanPak, and TanTel are trademarks of Compukit Corporation.

HARD DRIVE SPECIALIST

Ordering Information
Use our Watts line to place your order Via Visa, MasterCard, or Wire Transfer. Or Mail you payment directly to us. Any non-certified funds will be held until proper clearance is made. COD orders are accepted as well as purchase orders from government agencies. Most items are shipped off the shelf with the exception of hard drive products that are custom built. UPS ground is our standard means of shipping unless otherwise specified. Shipping costs are available upon request.

1-713-480-6000
Order Line 1-800-231-6671
16208 Hickory Knoll,
Houston, Texas 77059

Using *BAREAD 2.1*

Bar code listings must be read in numerical order beginning with Line 1 and continuing through the last line of the listing. The computer display is used to prompt you as to which line to scan and give you warning messages should you happen to get out of step.

When you run *BAREAD*, it asks you to scan the first line of the bar code listing. This line contains the name of the program as well as the beginning of the program itself. The computer will sound a high-pitched beep whenever it's ready for you to scan a line. After a line has been successfully read, you'll hear a lower beep. A "blip-bloop" sound prompts you to turn your attention to the screen for a message. You'll hear this when you accidentally scan a line out of sequence.

After reading the first line, you continue scanning with the second line. Remember to wait for a high beep before scanning and then listen for a low beep to indicate a successful read.

Once the last line of the listing has been scanned, *BAREAD* will return control to the Tandy 100/200 menu screen. Note that the program you just scanned is now in the directory with a `.DO` extension.

The final step is to convert the `.DO` text file to a normal BASIC program. This is done quite simply by going to BASIC and loading the file with a command such as `LOAD"TEST.DO"` (if the program name were `TEST`). The program will load into BASIC and will be ready to run. To save the program in BASIC's compressed format (`.BA` extension), you'd type `SAVE"TEST"` (if the program were named `TEST`). You may then kill the `.DO` file with `KILL "TEST.DO"`.

Bar Code Update

In case you missed this message last month, we would like to repeat it here. There has been a small change made to the PCM Bar Code Reader program in Line 1020. Without this change, you may get a ?BN error when trying to read code — especially after running some other programs which set the maximum number of files lower than the two required by the PCM Bar Code Reader. Change Line 1020 of your copy of the program to read as follows. This change has also been incorporated into the new listing in this issue.

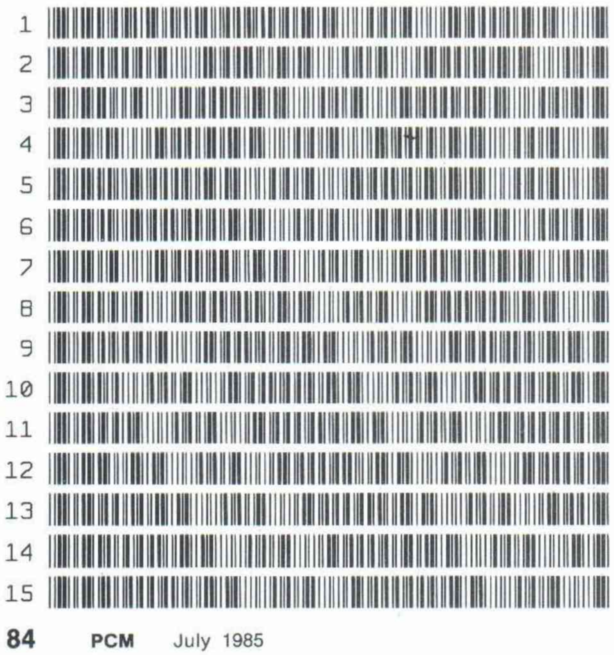
```
1020 CLEAR 1000:MAXFILES=2
```

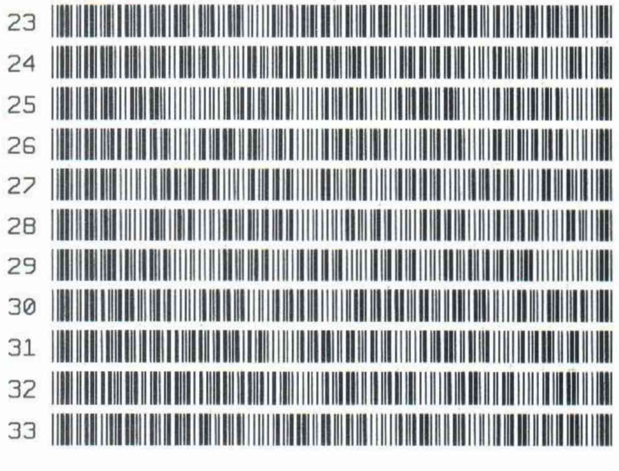
BAREAD Version 2.1

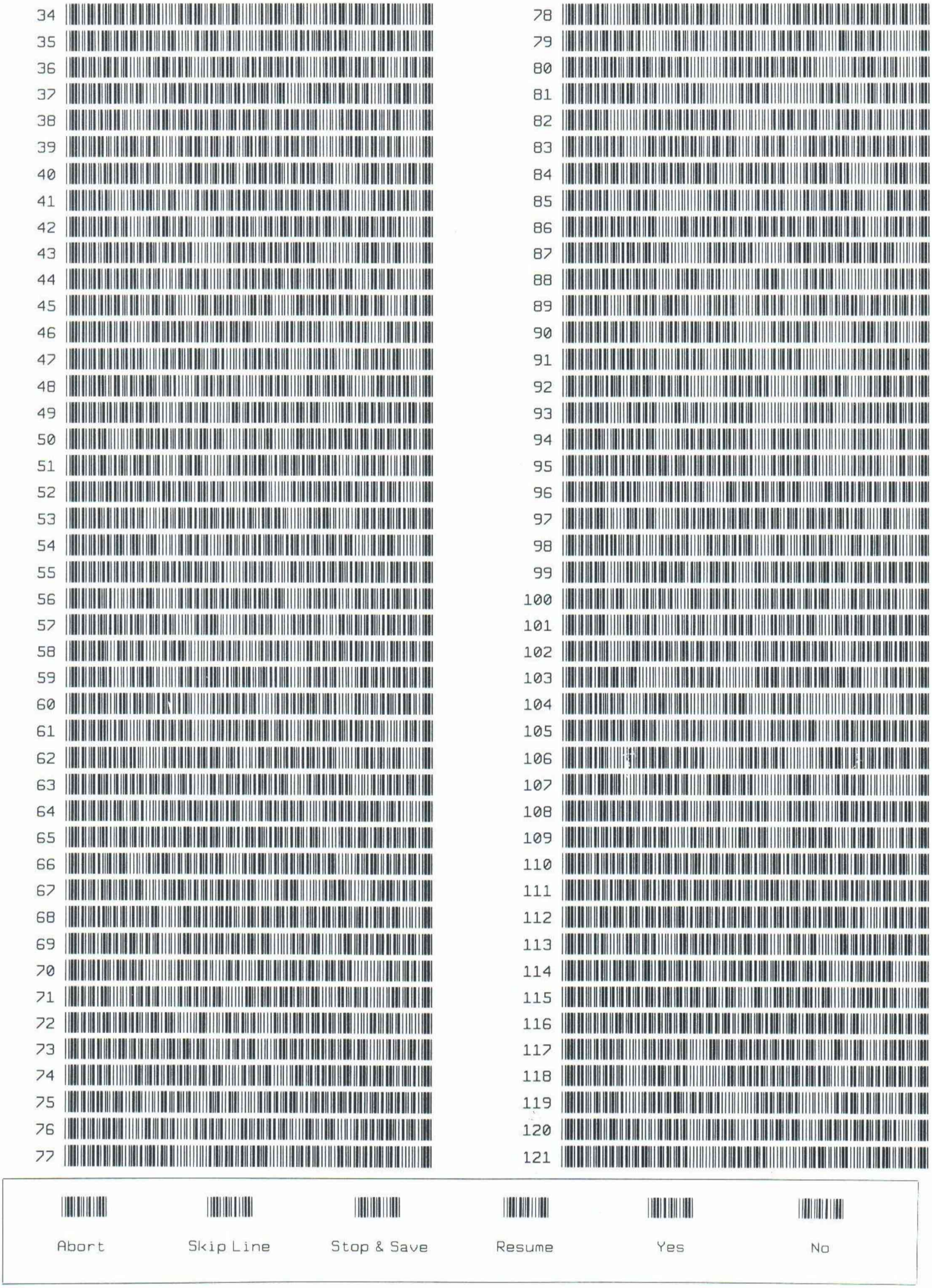
```
1000 ' *** Initialize ***
1010 ON ERROR GOTO 1040
1020 CLEAR 1000:MAXFILES=2
1030 GOTO 1050
1040 IF ERR=5 THEN RESUME NEXT
1050 ON ERROR GOTO 0
1060 RUNM "B30F9"
1070 OPEN "WAND:" FOR INPUT AS #1
1080 UC%=-1
1090 PC$="0123456789ABCDEFGHIJKLMNOPQRST
UVWXYZabcdefghijklmnopqrstuvwxyz- $+"
1100 DIM RW$(36)
1110 ER$(1)="You must scan line 1 first!
```

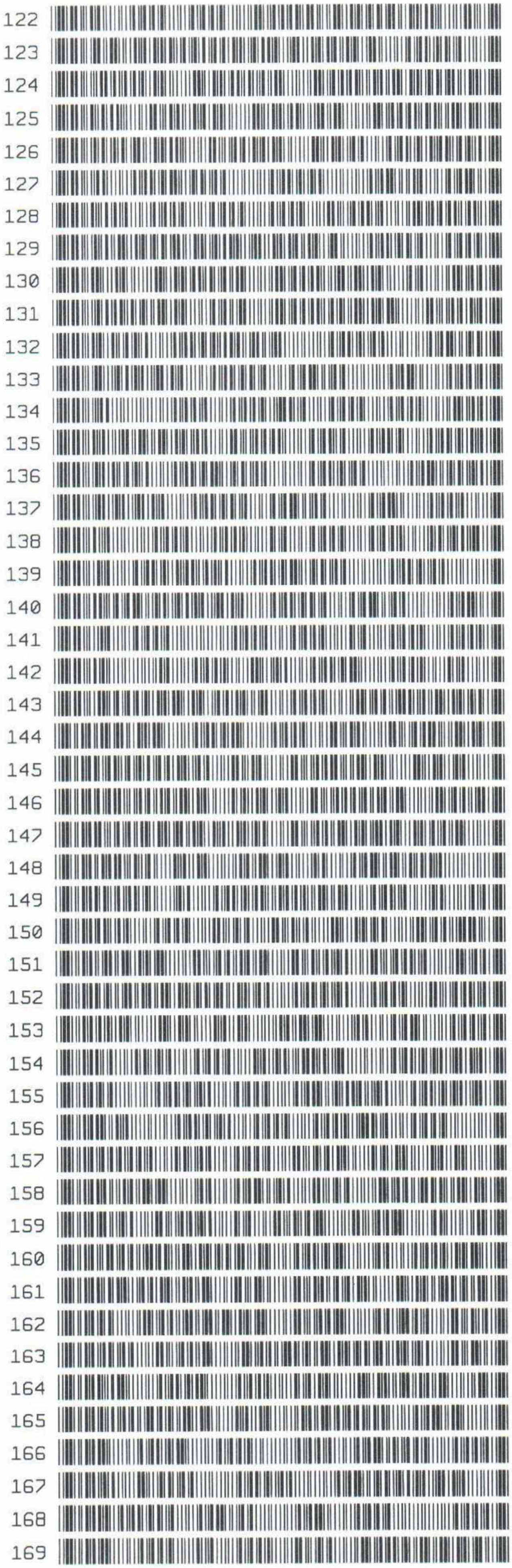
```
"
1120 ER$(2)="You've SKIPPED a line!"
1130 ER$(3)="You've ALREADY SCANNED this
 line!"
1140 ER$(4)="Code not PCM2/39 format!"
1150 ER$(5)="Command not applicable here
!"
1160 ER$(6)="You cannot skip this line!"
1170 ER$(7)="Selected resume file not in
 computer!"
1180 ' *** Read Reserved Words List ***
1190 DATA BEEP,CLEAR,CLOSE,DATA,DEFDBL,D
EFINT,DEFNG,DEFSTR,ELSE,GOSUB,GOTO
1200 DATA INKEY$,INPUT,INSTR(,LCOPY,LEFT
$(,LINE(,LOADM,LPRINT,USING,MAXFILES
1210 DATA MID$(,NEXT,PEEK,POKE,POWER,PRE
SET(,PRINT,READ,RESTORE,RETURN,RIGHT$(
1220 DATA SOUND,SPACE$(,STRING$(,THEN
1230 FOR I%=1 TO 36:READ RW$(I%):NEXT I%
1240 ' *** Procedure Begins Here ***
1250 CLS:PRINT@44,"PCM Bar Code Program
Reader v2.1"
1260 LINE(20,4)-(219,18),1,B:LINE(22,6)-
(217,16),1,B
1270 NN%=1
1280 GOSUB 1660:IF ER%>0 THEN GOSUB 1620
:GOTO 1280
1290 IF LL%=0 AND INSTR("YN",IL$)>0 THEN
 ER%=5:GOSUB 1620:GOTO 1280
1300 IF LL%=0 THEN ON INSTR("ALSR",IL$)
GOTO 1820,1890,1980,2050
1310 IF LL%=1295 THEN 1350
1320 IF LL%<>NN% AND NN%=1 THEN ER%=1:GO
SUB 1620:GOTO 1280
1330 IF LL%<NN% THEN ER%=3:GOSUB 1620:GO
TO 1280
1340 IF LL%>NN% AND NN%>1 THEN ER%=2:GOS
UB 1620:GOTO 1280
1350 IL$=RIGHT$(IL$,19)
1360 IF LL%=1 AND NN%>0 THEN GOSUB 1780
1370 CL$=CL$+IL$
1380 FOR I%=1 TO LEN(CL$)
1390      CH$=MID$(CL$,I%,1)
1400      IF CH$="%" THEN GOSUB 1510:IF NL
% THEN 1470 ELSE GOTO 1440
1410      IF CH$="/" THEN GOSUB 1550:IF NL
% THEN 1470 ELSE GOTO 1440
1420      IF CH$="." THEN UC%=NOT(UC%):GOT
O 1450
1430      IF CH$=>"A" AND CH$<="Z" AND NOT
(UC%) THEN CH$=CHR$(ASC(CH$)+32)
1440      XX$=XX$+CH$:IF RIGHT$(XX$,1)=CHR
$(13) THEN PRINT#2,XX$;:XX$="":UC%=-1
1450 NEXT I%
1460 CL$=""
1470 PRINT@200,SPACE$(80);
1480 IF LL%<>1295 THEN NN%=LL%+1:GOTO 12
80
1490 ' *** Done ***
1500 CLOSE:CALL 61807!:CLEAR 500,HIMEM:M
ENU
1510 ' *** Decode Reserved Word ***
1520 NL%=0:IF I%>LEN(CL$)-1 THEN NL%=-1:
CL$="%":GOTO 1540
1530 I%=I%+1:CH$=RW$(INSTR(PC$,MID$(CL$,
I%,1)))
1540 RETURN
1550 ' *** Decode Hex and Control Charac
ters ***
1560 NL%=0:IF I%>LEN(CL$)-1 THEN NL%=-1:
CL$="/":GOTO 1610
1570 I%=I%+1:IF INSTR("/%.",MID$(CL$,I%,
1))>0 THEN CH$=MID$(CL$,I%,1):GOTO 1610
1580 IF I%>LEN(CL$)-1 THEN NL%=-1:CL$=RI
GHT$(CL$,2):GOTO 1610
1590 HX$=MID$(CL$,I%,2):CH$=CHR$((INSTR(
"0123456789ABCDEF",LEFT$(HX$,1))-1)*16+I
NSTR("0123456789ABCDEF",RIGHT$(HX$,1))-1
)
1600 I%=I%+1
1610 RETURN
1620 ' *** Error Codes ***
1630 SOUND 5000,10:SOUND 8000,10:SOUND 5
000,10
1640 PRINT@220-.5*LEN(ER$(ER%)),ER$(ER%)
;
1650 RETURN
1660 ' *** Get Code Line ***
1670 PRINT@173,"";:PRINT USING "Scan lin
e ###";NN%
1680 IF NN%=-1 THEN PRINT@173,"Scan any
line":GOTO 1700
1690 SOUND 500,5
1700 INPUT#1,IL$:ER%=0
1710 FOR I%=1 TO LEN(IL$)
1720 IF MID$(IL$,I%,1)="!" THEN MID$(IL$
,I%,1)="."
1730 NEXT I%
1740 IF LEN(IL$)<>1 AND LEN(IL$)<>21 THE
N ER%=4:RETURN
1750 IF LEN(IL$)=1 THEN LL%=0:RETURN
1760 LL$=LEFT$(IL$,2):LL%=(INSTR("012345
6789ABCDEFGHIJKLMNOPQRSTUVWXYZ",LEFT$(LL
$,1))-1)*36+INSTR("0123456789ABCDEFGHIJK
LMNOPQRSTUVWXYZ",RIGHT$(LL$,1))-1
1770 RETURN
1780 ' *** Open Program File ***
1790 PN$=LEFT$(IL$,6):IL$=RIGHT$(IL$,LEN
(IL$)-6)
1800 OPEN PN$ FOR OUTPUT AS #2
1810 RETURN
1820   *** Abort ***
1830 BEEP:BEEP:BEEP
1840 PRINT@209,"ABORT!  Are you sure?";
1850 INPUT#1,AN$
1860 IF INSTR("YN",AN$)=0 THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 1850
1870 PRINT@200,SPACE$(80);
1880 IF AN$="Y" THEN CLOSE:KILL PN$+".DO
":GOTO 1490 ELSE GOTO 1280
1890 ' *** Skip Line ***
1900 IF NN%=1 THEN ER%=6:GOSUB 1620:GOTO
 1280
1910 BEEP:BEEP:BEEP
1920 PRINT@210,"SKIP!  Are you sure?"
1930 INPUT#1,AN$
1940 IF INSTR("YN",AN$)=0 THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 1930
1950 PRINT@200,SPACE$(80);
1960 IF AN$="Y" THEN NN%=NN%+1
1970 GOTO 1280
1980 ' *** Stop & Save ***
1990 BEEP:BEEP:BEEP
2000 PRINT@207,"STOP & SAVE! Are you sur
e?";
2010 INPUT#1,AN$
2020 IF INSTR("YN",AN$)=0 THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 2010
2030 PRINT@200,SPACE$(80);
2040 IF AN$="Y" THEN 1490 ELSE GOTO 1280
2050 ' *** Resume ***
2060 IF NN%<>1 THEN ER%=5:GOSUB 1620:GOT
O 1280
2070 PRINT@254,"Resume Mode";
2080 NN%=1:GOSUB 1660
2090 IF LL%=0 THEN ER%=5 ELSE IF LL%<>1
THEN ER%=1
2100 IF ER%>0 THEN GOSUB 1620:GOTO 2060
2110 PN$=MID$(IL$,3,6)
2120 ON ERROR GOTO 2140
2130 OPEN PN$ FOR INPUT AS #2:GOTO 2170
2140 RESUME 2150
2150 CLOSE #2
2160 ER%=7:GOSUB 1620:GOTO 1270
2170 CLOSE #2:OPEN PN$ FOR APPEND AS #2
2180 NN%=-1:GOTO 1280
```

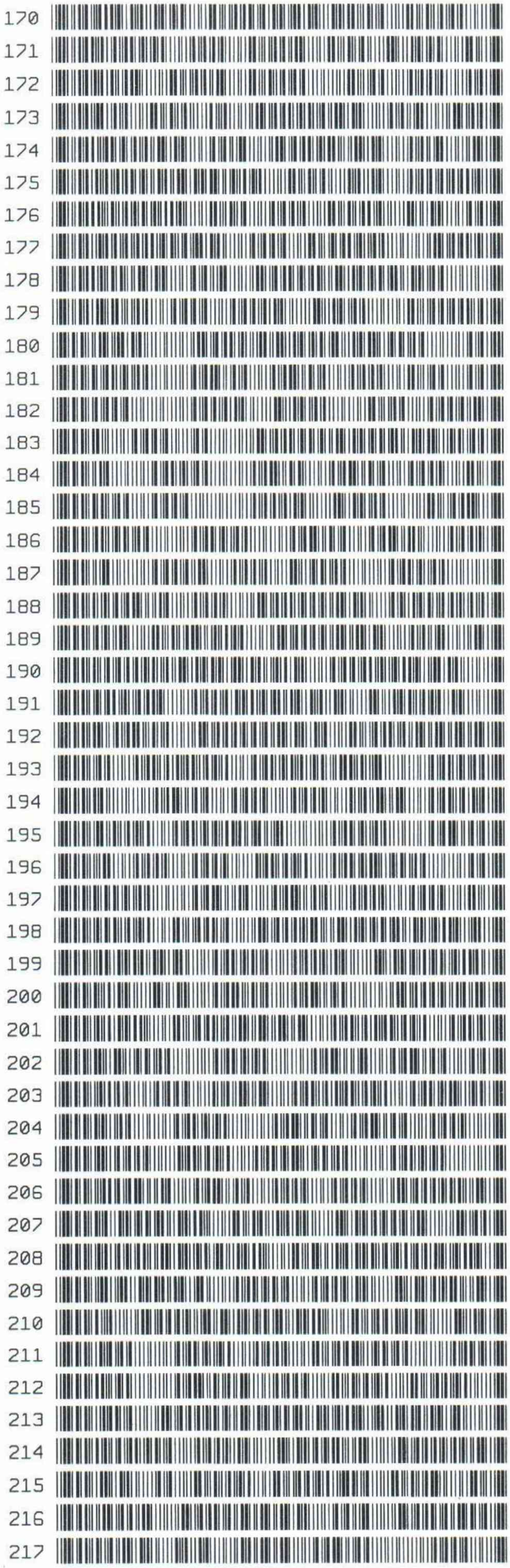
SCHEDL (FROM PAGE 31)

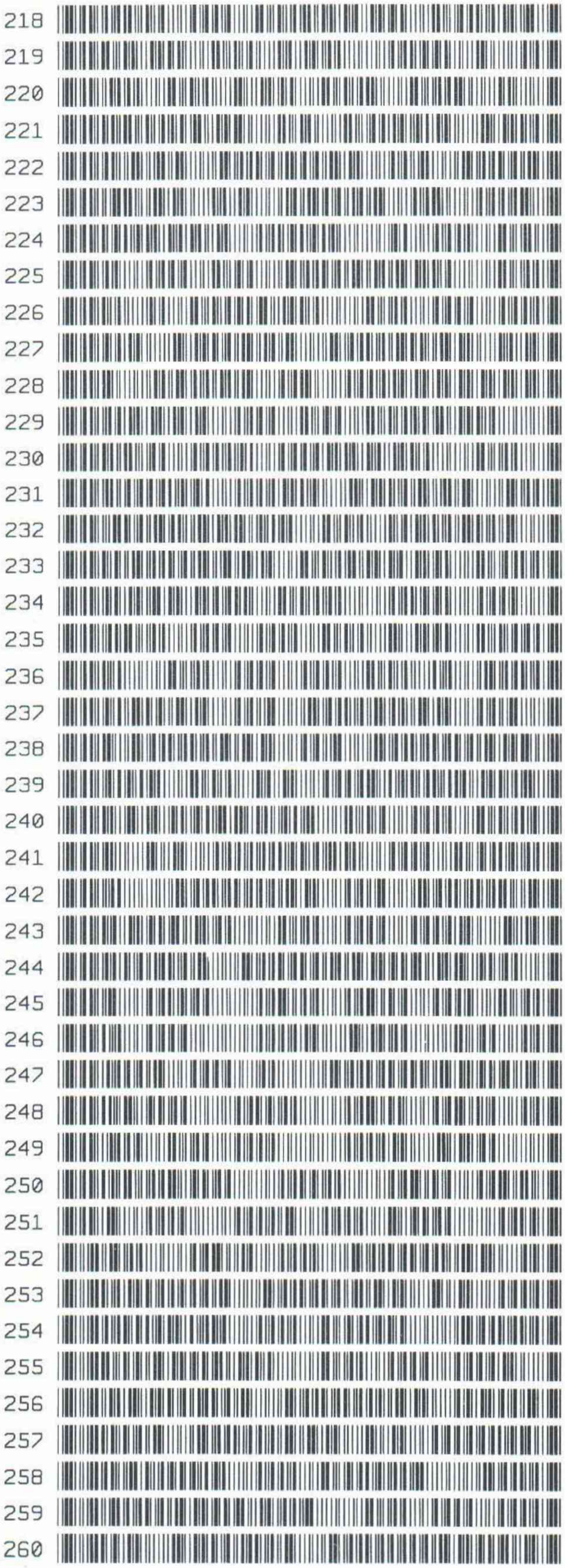


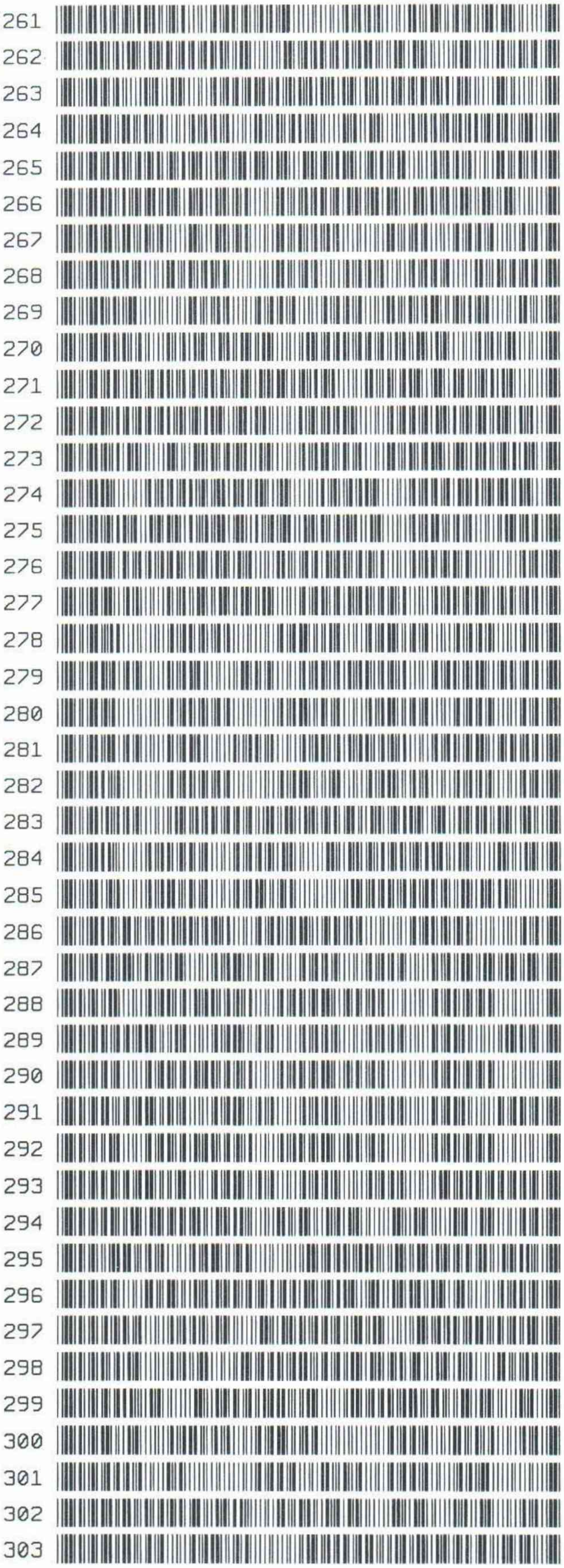


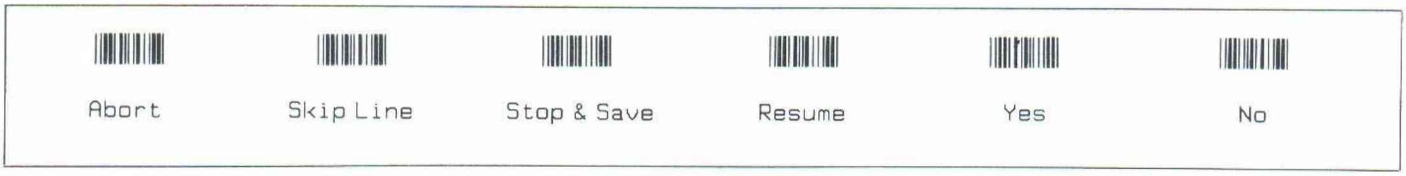

34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
Abort
Skip Line
Stop & Save
Resume
Yes
No

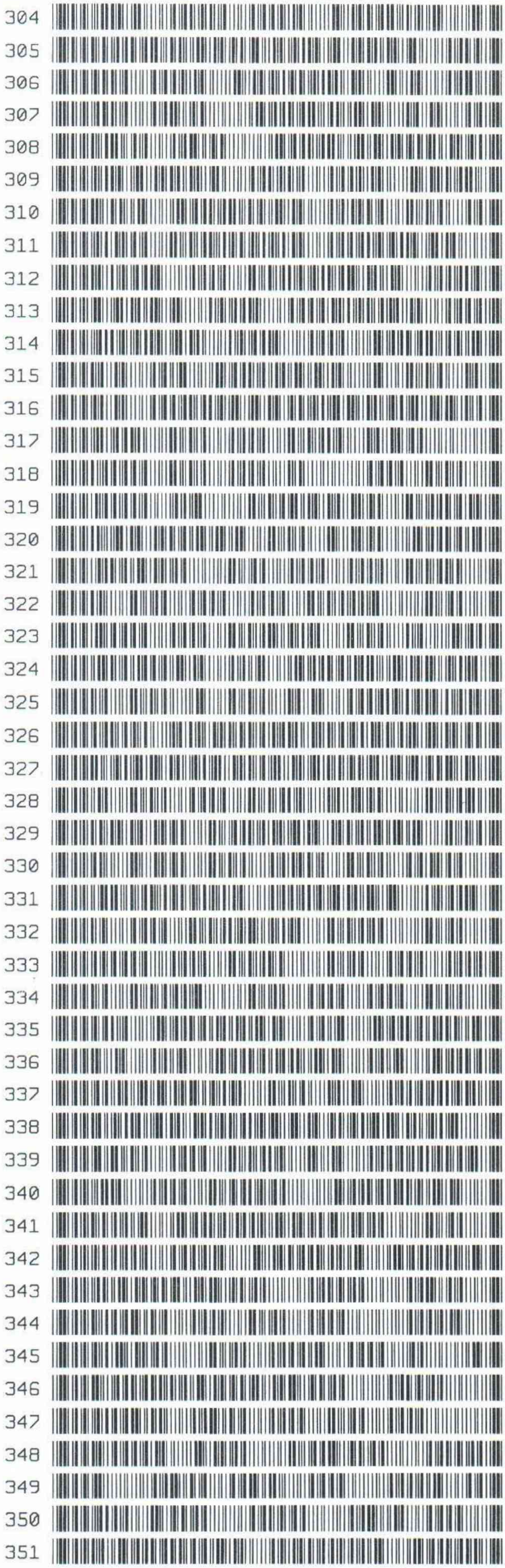

122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169

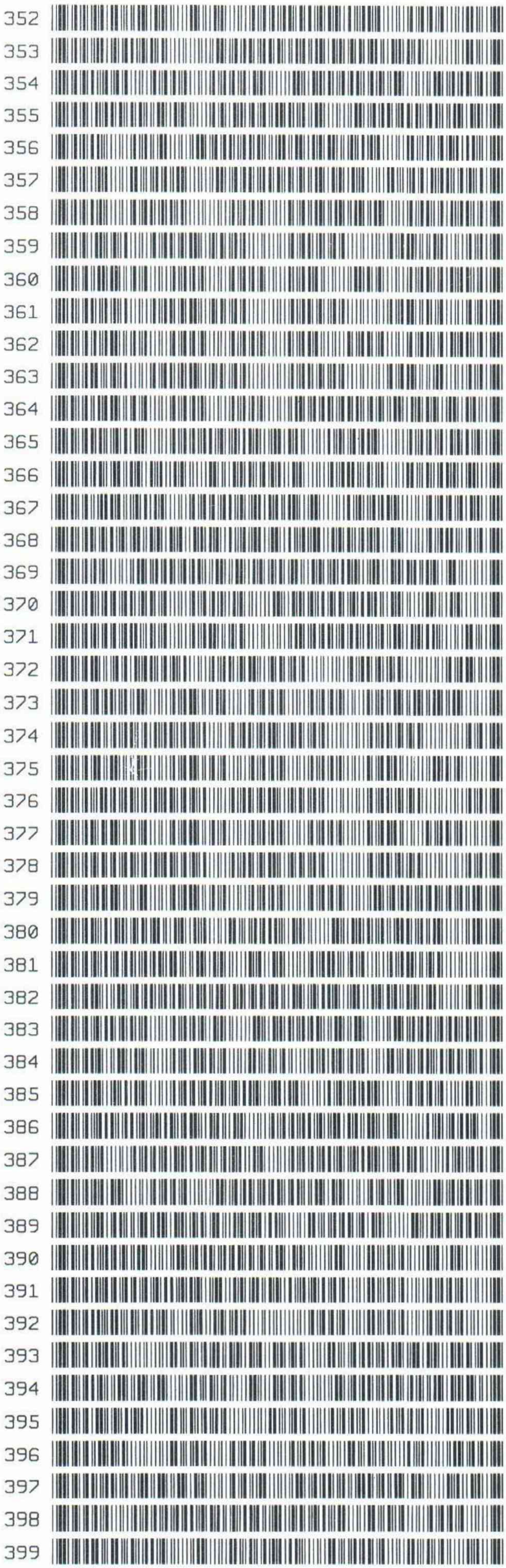

170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217


218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260

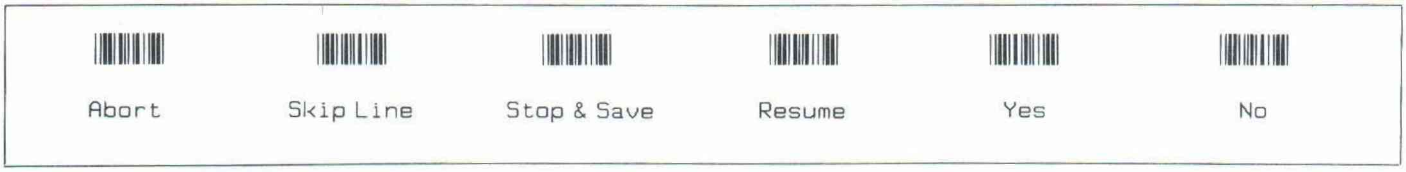

261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303


Abort
Skip Line
Stop & Save
Resume
Yes
No


304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351


352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485


Abort
Skip Line
Stop & Save
Resume
Yes
No

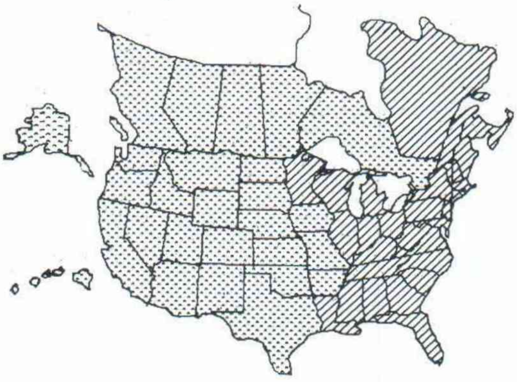
The Rackseller

ALABAMA	
Birmingham	Jefferson News Co.
Florence	Anderson News Co.
Madison	Madison Books
ALASKA	
Fairbanks	Fairbanks News Agency
ARKANSAS	
Fayetville	Vaughn Electronics/Radio Shack
ARIZONA	
Phoenix	Computer Pro
	Tri-Tek
Scottsdale	Softwareland Corp.
Tucson	Anderson News Co.
CALIFORNIA	
Citrus Heights	Software Plus
Half Moon Bay	Strawflower Electronics
Livermore	Software Galeria
Lompoc	L & H Electronics Emporium
North Hollywood	Levity Distributors
San Francisco	News on 24
Santa Rosa	Sawyer's News
Sunnyvale	Computer Literacy
DELAWARE	
Middletown	Delmar Co.
Wilmington	Normar Inc.—The Smoke Shop
FLORIDA	
Altamonte Springs	International Music & Recordings
Boca Raton	Software, Software, Inc.
Ft. Lauderdale	Software Connection
N. Miami Beach	Almar Bookstore
Panama City	Computet Systems Group
Pensacola	Anderson News Co.
Tallahassee	Anderson News Co.
GEORGIA	
Atlanta	Guild News Agency
Columbus	Muscogee News Co.
ILLINOIS	
Chicago	Chas. Levy Distr.
	Kroch's & Brentano's Jackson Street
	Kroch's & Brentano's Wabash Ave.
	Prairie News Agency
Oakbrook	Kroch's & Brentano's
INDIANA	
Elkhart	Elkhart City News & Book Store
Mishawaka	Carrico's Radio Shack
Scottsburg	Radio Shack of Scottsburg
IOWA	
Ames	Myers TV & Stereo Repair
Davenport	Interstate Book Store
KANSAS	
Wichita	Lloyd's Radio
KENTUCKY	
Louisville	Hawley-Cooke Bookstores
Prospect	Falsoft, Inc.
LOUISIANA	
Gretna	Computer Supply Store
Slidell	Radio Shack
MARYLAND	
Laurel	Pete Brewin
Lexington Park	Books, Etc.
MASSACHUSETTS	
Littleton	Computer Plus
Marlboro	Radio Shack
MICHIGAN	
Rochester	Rochester Book Center
Sterling Heights	Software City Programs, Inc.
Wyoming	Gerry's Book Co.
MISSOURI	
Independence	Digital Enterprises, Inc.
NEBRASKA	
Lincoln	Hobby Town
NEW HAMPSHIRE	
Manchester	Brookwrights
Petersborough	BRW Electronics-Radio Shack
West Lebanon	Verham News Corp.
NEW JERSEY	
Clinton	Micro World II
Hackensack	Total Circulation Service
Marmora	Outpost Radio Shack
Pennsville	Dave's Electronic Radio Shack
NEW MEXICO	
Alamogardo	New Horizons Computer Systems
Albuquerque	News and Bookstore
	Page One Newsstand
NEW YORK	
Hudson Falls	G.A. West & Co.
NORTH CAROLINA	
Cary	It's Just For You, Inc.
	News Center in Cary Village
Sylva	Total Data Systems
NORTH DAKOTA	
Fargo	Computer Associates
OHIO	
Cincinnati	Cinsoft
Toledo	Leo's
OKLAHOMA	
Oklahoma City	Merit Computers
PENNSYLVANIA	
Pleasant Hills	Pittsburgh Computer Store
RHODE ISLAND	
Newport	Kelly's Variety
SOUTH CAROLINA	
North Charleston	The Green Dragon
TENNESSEE	
Knoxville	Anderson News Co.
Memphis	Computer Center
Nashville	Mosko's Book Store
Smyrna	Delker Electronics, Inc.
TEXAS	
Dallas	Micro Concepts, Inc.
Ft. Worth	A & A International
VIRGINIA	
Alexandria	Alonso Book & Period
Norfolk	I-O Computers
WASHINGTON	
Kennewick	Satellite TV Specialists
WISCONSIN	
Ladysmith	Electronics, Etc.
CANADA:	
ALBERTA	
Calgary	Rainbow Software Services
Edmonton	CMD Micro
	Kelly Software Distributors
BRITISH COLUMBIA	
Sidney	Sidney Electronics
QUEBEC	
LaSalle	Messageries De Presse Benjamin
NOVA SCOTIA	
Halifax	Atlantic News

Also available at selected B.Dalton Bookseller stores

Advertisers Index

PCM Advertising and Marketing Representatives



▒

Cindy Shackleford
Director, West Coast Office
12110 Meridian South — Suite 8
P.O. Box 73-578
Puyallup, WA 98373-0578
(206) 848-7766

▨

Kate Tucci
Advertising Representative
9529 U.S. Highway 42
P.O. Box 385
Prospect, KY 40059
(502) 228-4492

Join us for our very first show . . .

PCMfest℠

Princeton, N.J. Oct. 11-13

You're invited to PCMfest, a great new show that's just for your Tandy computer. Sponsored by PCM, *The Personal Computing Magazine for Tandy Computer Users*, it's a wonderful way to meet and exchange information with those who share your interest in the new generation of Tandy computers.

You'll also discover the greatest variety of products ever for your computer because all of the favorites and a lot of new ones will be featured in the exhibits of PCM advertisers. Try out that new program and take it home that very day!

Along with other PCM readers, you'll meet the top national experts on your computer, including those who write for or who are written about in PCM. They will answer your questions on the spot.

PCMfest also will include a comprehensive lineup of free seminars on topics of immediate concern — and all of them designed to help you get the most out of your Tandy computer.

The Hyatt Regency—Princeton will be offering special rates ($65, single or double room) for PCMfest. The show opens Friday evening with a 7 p.m. to 10 p.m. session. It's a daytime-only show Saturday — the exhibits open at 10 a.m. and run continuously until 6 p.m. On Sunday, the exhibit hall opens at 11 a.m. and closes at 4 p.m.

Tickets may be obtained directly from PCM. We'll also send you a special reservation form so you can take advantage of the special room rate. Come to PCMfest and let's celebrate the new generation of Tandy computers!

RAINBOWfest

Your admission to PCMfest also entitles you to attend RAINBOWfest, the highly popular show for the Tandy Color Computer which will run concurrently with PCMfest at the same location. The shows are hosted by Falsoft, Inc., publisher of both PCM and THE RAINBOW, *The Color Computer Monthly Magazine*.

RAINBOWfest will feature a great variety of exhibits, new products, demonstrations, and free seminars for the Tandy Color Computer.

YES, I'm coming to Princeton! I want to save by buying tickets now at the special advance sale price.

Please send me:

_____ three-day tickets at $9 each total _______

_____ one-day tickets at $7 each total _______

(Circle one) Friday Saturday Sunday

Handling Charge $1 $1.00

TOTAL ENCLOSED _______

U.S. FUNDS ONLY, PLEASE

□ Also send me a hotel reservation card for the Hyatt Regency—Princeton, ($65, single or double room).

Name (please print) _______________

Address _______________

City _______________ State _______

Telephone _______________ ZIP_______

Company _______________

Charge to:
□ VISA □ MasterCard □ American Express

Account Number _______________

Exp. Date _______________

Signature _______________

Make checks payable to: PCM. Mail to: PCMfest, the Falsoft Building, P.O. Box 385, Prospect, KY 40059. To make reservations by phone, call: (502) 228-4492.

Advance ticket deadline: October 4, 1985. Orders received less than two weeks prior to show opening will be held for you at the door. Tickets will also be available at the door at a slightly higher price.

The Chipmunk™ A Revolutionary Portable Disk Drive to Accompany Your Model 100

. . . for only $599

Includes Six Dynamic Programs.

Holmes/Amtek introduces the Chipmunk,™ a highly sophisticated portable disk drive that allows you to access 358K of a 3½" diskette.

Imagine the storage capacity and speed of a disk drive in your briefcase, ready to go to work for you — anytime, anywhere.

The CDOS operating system is function key driven, versatile, and easy to learn and apply. Whether it's word processing, spreadsheets, data bases, or applications you write yourself, our Chipmunk is a friendly, industrious little fellow.

SPECIFICATIONS

Height: 2 inches; **Width:** 5½ inches; **Depth:** 7½ inches; **Weight:** 2½ pounds; **Case:** ABS plastic; **Interface:** Parallel via expansion bus with EZ-Snap ribbon cable connectors; **Transfer:** 10 Kilobytes / second effective speed, 150,000 Baud actual rate; **Disk Media:** 3.5" SSDD 80-track micro floppy diskettes, 358.5K formatted; **Power:** Internal, rechargeable NiCad batteries or AC charger / adapter; **Battery Life:** 2 + hrs. constant disk I/O, more than 10-12 hrs. in normal use; **DOS Load:** CDOS loads from Chipmunk ROM w/o cassette or program; **Features:** Familiar menu displays with multiple "folders" for storing up to 239 files and/or folders per disk. Enhanced function keys and familiar use of arrow keys make CDOS easy to learn and use. "Save All," "Load All," "Name," "Kill" with function keys plus date flagging of disk files and on-screen "Help" program. **Price:** $599 complete with adapter module, cable, System Disk, AC charger/adapter and manual. Includes six excellent programs.

Portable Computer Support Group, Inc.
11035 Harry Hines Blvd., #207
Dallas, Tx. 75229
1-214-351-0564
MC/VISA/COD

CALL US FOR 8K RAM UPGRADE CHIPS, $49.95